目　录

第一章　电脑游戏机系统概论

随着“任天堂”系列电视游戏机的迅速普及，电脑游戏机已逐步进入我国家庭，大有发起我国玩具市场的第四次革命之势（第一次是匈牙利的魔方、第二次是美国的变形金刚、第三次是日本的“任天堂”系列家庭电视游戏机）。这种电脑学习机充分利用了我国家庭已经普及的电视机、录音机和正迅速普及的“任天堂”系列游戏机，仅仅配置输入设备——键盘和学习卡，就组成了一套较完整的家庭电脑系统。它既是一种高档的电子玩具，又是中、小学生的家庭辅导教师；既能引发少年儿童学习电子计算机的兴趣，又能提高儿童的智商和逻辑思维能力。因而，它以其较高的性能价格比、较全面的实用功能，越来越受到广大家长的重视。尤其是它的谱曲演奏、算术运算、键控绘图、英文打字、F BASIC 语言编程五大基本功能已深深地吸引了广大青少年朋友，成为他们工作学习中的良师益友，生活娱乐中的亲密伙伴。

目前，市场上又逐渐推出了普里奇声像学习系统、浮点 BASIC 工作系统、LOGO 工作系统和英语学习、中文打字专家、中小学辅助教学等软件；尚有家庭健康咨询、缝纫烹调、卡拉 OK 字幕系统、任天堂游戏开发系统等软件正在紧锣密鼓的开发中；硬件结构上已完善了打印机、软盘驱动器、101 标准键盘等外设接口。这些，更使电脑游戏机如虎添翼。而且有些产品，如裕兴键盘采取了以 DIR 命令列出文件目录、以 EXEC 命令进入选定文件的操作方式，引导用户在电脑学习机的使用中初步建立了 PC 机中文件的概念，从而真正起到了向 PC 机过渡的桥梁作用，进一步缩短了电脑游戏机与 PC 机的距离。相信随着电脑游戏机的推广普及，将会有更多更好的工作系统和软件问世。电脑游戏机必将成为小至二、三岁幼儿、上至花甲老人的案头宠物；必将成为我国家庭不可少的家用电器；必将为人们的工作、学习、生活带来极大的方便、实惠和无穷的乐趣。

1·1　引　　言

几乎所有拥有电脑游戏机的朋友都提出了以下几个相同的问题：

1、F BASIC 的背景显示分辨率为 28 列×24 行，而任天堂游戏中的背景显示分辨率却为 32 列×30 行。相比之下，F BASIC 的显示画面仅为任天堂游戏画面的十分之七。因此，用 F BASIC 语言设计的游戏画面远不如任天堂游戏画面的内容丰富。既然 F BASIC 是一种面向游戏的语言，那么能否在 F BASIC 状态下编写显示 32 列×30 行的画面程序呢？

2、在大多数任天堂游戏中，如《魂斗罗》、《绿色兵团》等，音响、角色可以同步控制。即在背景音乐伴奏的同时，角色可以做着各种复杂的动作。但用 F BASIC 语言编写的游戏，音响的发出和角色的动作却是分步进行的，即在角色动作时没有音响；发出音响时角色的动作要停下来，直到音响结束后才能继续动作。那么，能否使用 F BASIC 语言编写音响、角色二者同步进行的程序呢？

3、在任天堂游戏中，游戏画面可以连续左右卷动或上下滚动。这种画面的卷动和滚动也可以和角色动作、音响同步进行。而用 F BASIC 语言编写的游戏程序却不能进行背景画面的卷动、滚动，只能一场一场的更换背景，从而使游戏单调、呆板。再如在《淘金者》二代的游戏中，每

一局的画面有 60 列×48 行，可通过方向键移动画面截取任一部分的 32 列×30 行的画面。既然任天堂游戏和 F BASIC 都用同一个 CPU，那么能否用 F BASIC 语言实现上述功能呢？

4、任天堂游戏中有着丰富逼真的效果音响。如飞机的升起降落、开枪、爆炸、开门、关门等声音。使用 F BASIC 语言虽然也能模仿部分效果音响，但总不如任天堂游戏中的逼真。尤其是模仿诸如警报声、引擎声等类音响，使用 PLAY 语句则很难实现。那么，在任天堂游戏中的这些音响是如何发出的？能否在 F BASIC 系统中实现呢？

5、F BASIC 系统中，卡通角色最大为 16×16 点阵，而任天堂游戏中的角色大多数是 32×32 点阵，有的甚至是 32×64 点阵。它们是怎样定义的？任天堂游戏中有各种各样的角色和背景，F BASIC 系统中为什么没有这么多？

这些问题一方面反映了朋友们强烈的求知欲望；另一方面也充分暴露了 F BASIC 语言的种种不足与缺陷。

为满足朋友们的愿望，弥补 F BASIC 语言目前存在的种种不足，本文拟从电脑游戏机的工作系统分析入手，简要介绍任天堂游戏机 6527CPU 工作系统的特点；重点分析电脑游戏机的人机对话工作系统和工作过程；讨论 F BASIC 工作系统中各指令的工作特点，从而找出使用 F BASIC 语言，以最小的程序量实现 32 列×30 行背景图形的绘制；背景画面的连续卷动和滚动；绘制 64 列×30 行背景画面；音响、角色同步工作以及发出各种效果音响等功能。并简要介绍各种卡通块、背景图形块的结构及定义方法，使读者能够使用 F BASIC 语言编写出与任天堂游戏相同效果的游戏程序片断，进而通过扩展内存，自己动手编写制作出完整的游戏软件。

为适应不同层次读者的不同要求，本书以拙作《电脑游戏机的 F BASIC 语言与编程技巧》和朱国江先生的《中华学习机机器语言》为基础进行编写。对要求全面掌握电脑游戏机工作系统的朋友，可先学习以上两本书后再阅读本书；对只想使用游戏编程特殊技巧的朋友则可直接阅读本书的第五章，只要遵照书中介绍的方法，把各子程序直接移植到自己编写的游戏程序中，必将小试有成，收到满意的效果

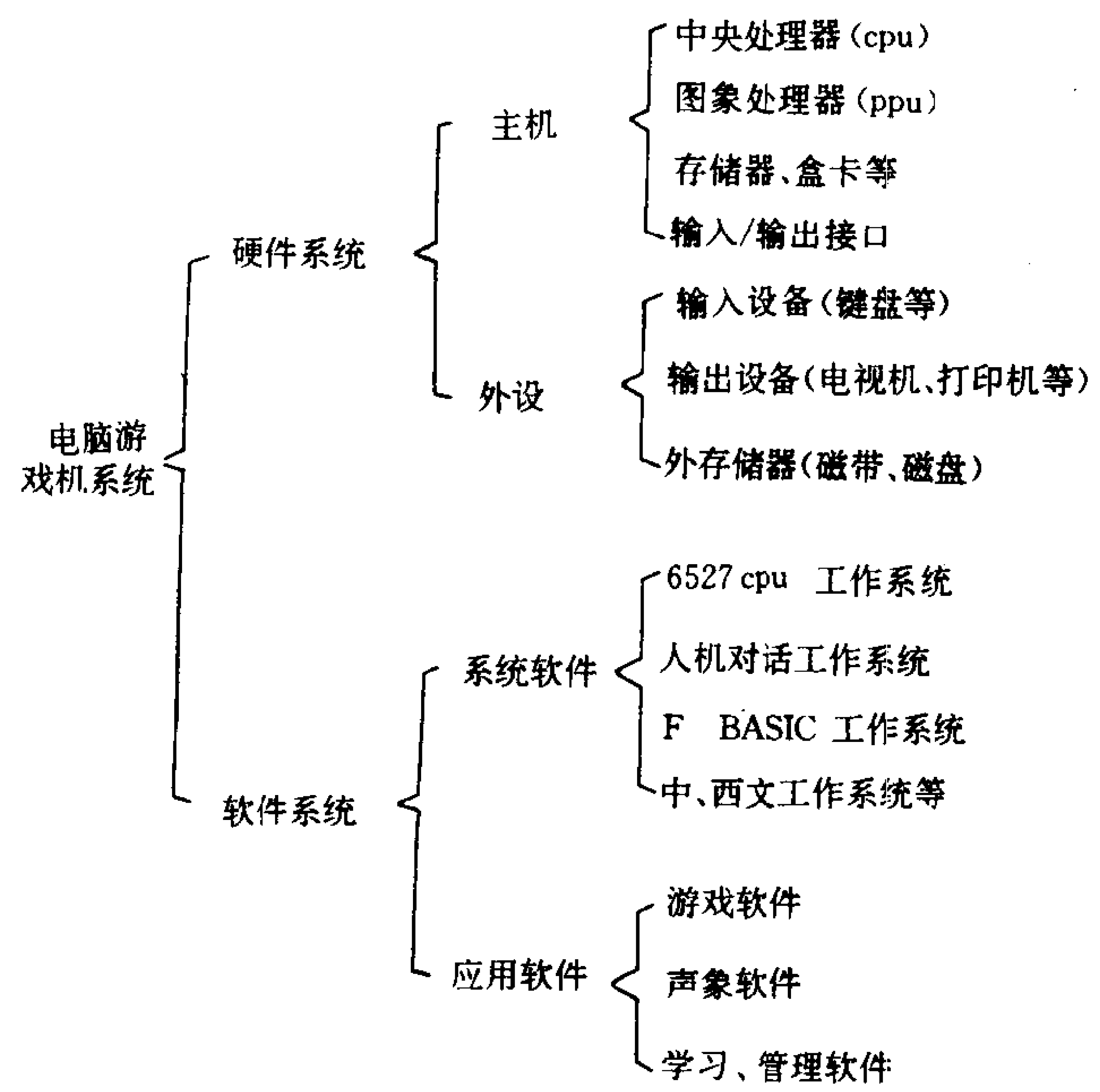


图 1-1 电脑游戏机的系统组成

1·2 电脑游戏机的系统组成

同所有的电子计算机一样，电脑游戏机系统也由硬件系统、软件系统两大部分组成。其结构见图1—1。

1·3 电脑游戏机的硬件系统

1·3·1 电脑游戏机的主机组成

目前流行的电脑游戏机，概括起来可分成两大类：分立式主机结构；集合式主机结构。

·分立式主机结构由任天堂游戏机(裸机)、键盘、学习卡(BS卡)三部分组成，见图1—2。

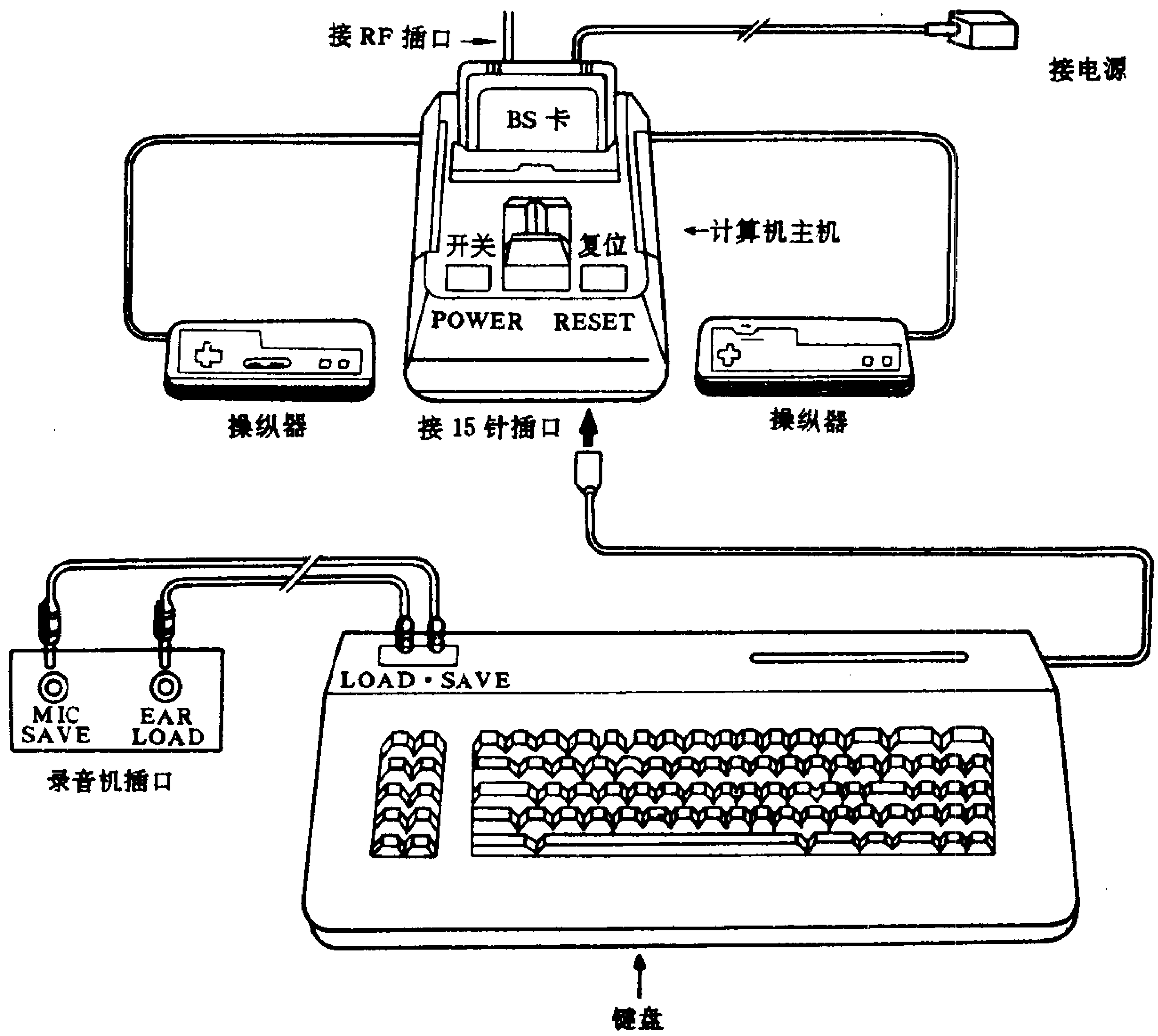


图1—2 分立式电脑游戏机的主机结构

分立式结构属积木式结构，用户可分级购买，可充分利用用户手中的游戏机，相对初次投资较小。另外，由于软件系统固化在可更换的BS卡中，故更换工作系统较方便。如北京裕兴机电研究所推出的超级学习卡专门设计了一个32针的扩展插槽，在该插槽上可以插入五笔字形、LOGO语言、英语学习、升学辅导、打字专家、任天堂游戏等各种应用软件芯片，使用户可以根据自己的实际需要选购不同的芯片，扩展电脑的功能。这样，既拓宽了用户的选择余地又节省了用户的一次性投资，故较受用户欢迎。因此，这种结构将成为游戏电脑领域最流行的结构。缺点是：主机、键盘分离，使用时要插接键盘，故使用不太方便。

· 集合式结构

集合式结构则把任天堂游戏机、键盘、学习卡集合于一体，把它们装在一个机壳内，见图1－3。

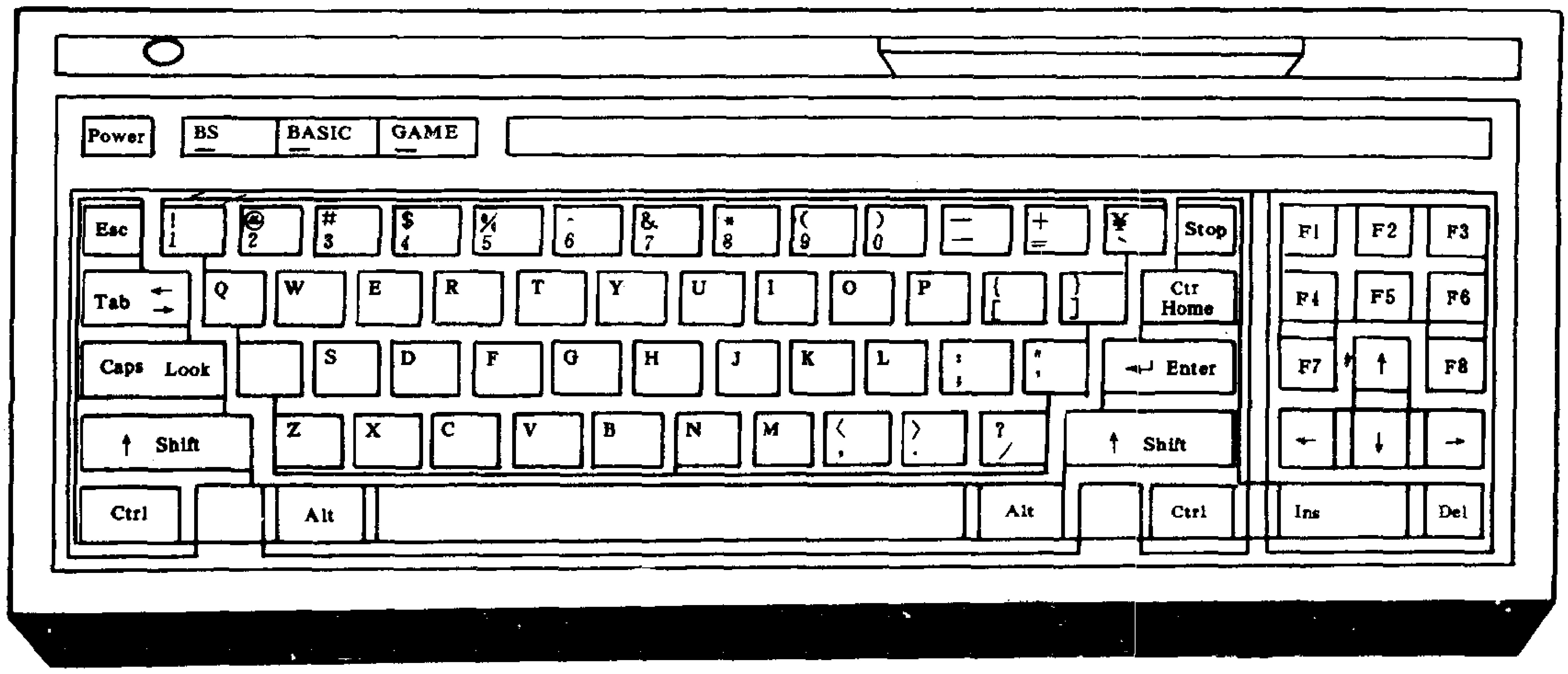


图1－3　集合式电脑游戏机主机

这种结构外形美观大方、富丽豪华。由于工作系统软件集中装在主机内，进入各个工作系统的转换操作极为简便，仅仅通过按键就可迅速转入目标系统。如在科特FCS－90电脑游戏机中，按"Ctrl＋Alt ＋F1"三键可进入人机对话工作系统和BS. 2A F BASIC工作系统；按"Ctrl＋Alt ＋F2"三键可进入V3. 10版本的F BASIC工作系统；按"Ctrl＋Alt ＋F3"三键可执行游戏卡中的游戏程序。各系统间的转换极为简便。另外，现在市场上推出的金字塔PEC－9388集合式电脑游戏机，机中配备了软盘驱动器接口，更方便了电脑游戏机的外存储操作。

这种结构的缺点是：一次性投资较大（比分离式结构要高出近50%）；工作系统已固定，用户无法更换或扩充。

不论那种主机结构，其硬件构成是基本相同的。即都由主控部分、接口电路、时钟电路、PAL制式射频调制器、键盘、电源和附件（操纵器、光电枪等）七部分构成。

一、主控部分

主控部分由CPU、PPU、ROM、RAM和门电路组成。

1、6527CPU——电脑游戏机的中央处理器，其英文名称为"Central Processing Unit"，缩写为CPU。它是一个八位的NMOS CPU，也是一种八位单片机。它具有八位数据线，十六位地址线，直接寻址64K空间，时钟频率为26. 601712MHZ，每秒钟可执行约200万次八位数据运算。使用单一＋5V电源。

2、6528PPU——电脑游戏机图象处理器，其英文名称为"Picture Processing Unit"，缩写为PPU它在电脑游戏中主要用于处理图象视频信号。它有八位数据线，十四位地址线，直接寻址16K空间，使用单一＋5V电源，时钟频率一般为21～27MHZ，故图象处理速度极快。

3、ROM——电脑游戏机的只读存储器，其英文名称为"Read Only Memory"。电脑游戏机中一般有两片ROM：一片为2764(8×8KB)，用于存储图形库数据（通称字模）；一片为27256(8×32KB)或27512(8×64KB)，用于存储工作系统软件。（中、西文学习卡中还将有一片或两片1M位的ROM固化汉字一、二级字库）

4、RAM——电脑游戏机的随机存取存储器，其英文名称为" Random Access Memory"。

电脑游戏机中一般有三片 RAM：两片 6116(8×2KB)分别充任 CPU 的中间数据存储器、PPU 的图象数据存储器；另一片为 6264(8×8KB)或 62256(8×32KB)用于存放用户程序。

5、40 系列、74 系列门电路若干块，分别充任计数、译码、反相、地址锁存、三态缓冲等作用，详见各部分介绍。

二、接口电路

从操纵器、光电枪、键盘等外设发出的外部控制信号都将通过接口电路挂接在 CPU 数据总线上。操纵器Ⅰ、Ⅱ分别通过游戏机电路板上的插座Ⅰ、Ⅱ与接口电路相连接；光电枪、键盘等则通过扩充插座与接口电路相连；打印机、软盘驱动器接口一般通过键盘接口与主机相连。

三、时钟电路

中央处理器、图象处理器以及接口电路等都必须按照一定的时间顺序执行相关的控制指令。因此，需要设置一个统一的时间标准，以协调整个系统的操作运行。这一统一的时间标准叫做时钟。产生这一时钟信号的电路就是时钟电路，通常叫做时钟脉冲发生器。

四、PAL 制式射频调制器

由于我国的彩色电视机信号标准是 PAL 制式的，所以为了适配国产彩电，要把由 PPU 输出的视频信号、CPU 输出的音频信号调制成 PAL－D 制式的射频信号，以供电视机接收、显示。

五、键盘

键盘是电脑游戏机的主要输入设备，软驱、打印机也都是通过键盘与主机相连接的。目前，市场上流行多种多样的键盘，早期的有小天才 IQ－90、天马 TM－5115、天津的 KB－40、科特 FCS－90 等；近期占市场主导地位的有裕兴、金字塔、小霸王、大岛、金钥匙等。近期产品大多数都支持中文汉字，因而给使用带来了更大的方便。

1·3·2　6527 中央处理器

电脑游戏机所使用的中央处理器——CPU，尽管因生产厂家不同、型号差异较大，但都属于 65 系列的微处理器。它们都有八条数据线、十六条地址线，具有 64K 寻址能力，采用单一的＋5V 电源。

一、任天堂、小天才系列游戏机的 CPU 型号对照情况见表 1－1。

表 1－1　电脑游戏机 CPU 型号对照表

机　型	CPU	时钟频率	相同功能的 CPU 型号
任天堂 天　马	6527 6527	21.251465MHz 或 21.47727MHz	87007、6005H、KD841、 2A03、KD840
小天才 胜天、科特	6527P 6527P	26.601712MHz 26.601712MHz	87007P、6540、2A03E、 P02、2B03、MG－P－502

二、6527CPU 各引脚功能简介

6527 与 6527P 除时钟频率不同外，其它都相同。各引脚名称见图 1－4。

各引脚功能如下：

- 1 脚、SOU1：第一模拟音频信号输出端。
- 2 脚、SOU2：第二模拟音频信号输出端。
- 3 脚、RESET：复位信号输入端，低电平有效。可令 CPU 结束当前的工作而返回到开机

时的状态。

• 4～19 脚、A0～A15:CPU 的 16 位地址线,由它组成地址总线,最大寻址能力为 64K 字节。地址总线在系统内对随机存储器 RAM、图象处理器 PPU、系统软件 ROM 进行寻址,以取出需要的信息进行运算处理、把运算结果送入 RAM 暂时保存、需要时再读出。

其中,加于 RAM 的地址线为 A0～A10 共 11 条,可寻址 2K 字节;A0～A2 作为 CPU 与 PPU 通信的地址线;A0～A14 共 15 根地址线作为 ROM 的输入地址线,可寻址 32K 字节,用于读取系统软件。

另外,CPU 的高位地址线 A13～A15 与 RDY 还作为双二线四线译码器(74LS139)的输入信号,经译码后转换成 1Y0、1Y1、2Y3 信号,作为片选信号分别选通 RAM、PPU、和 ROM。

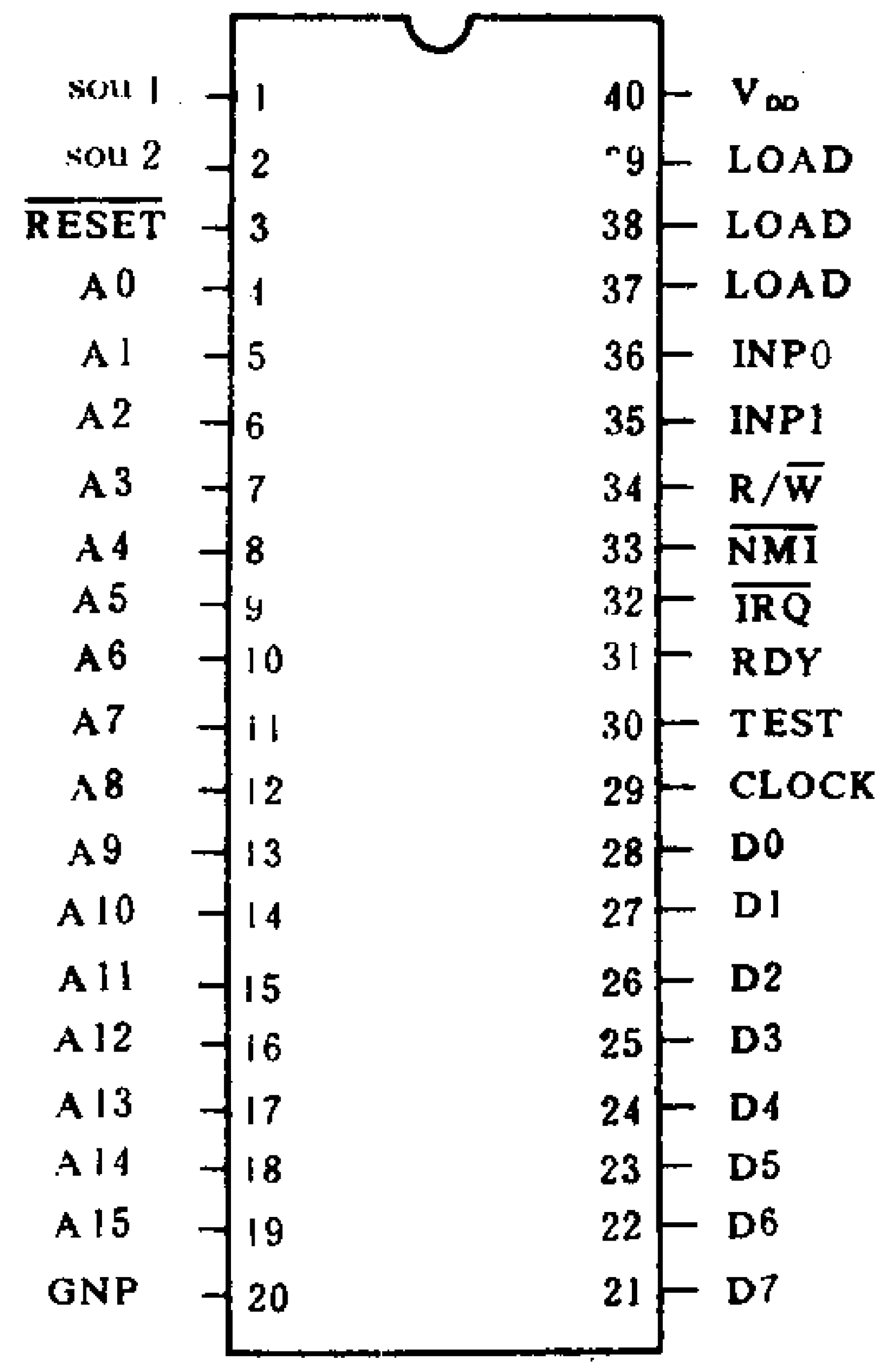


图 1－4　6527CPU 引脚排列图

• 20 脚、GND:地线端。

• 21～28 脚、D0～D7 是 CPU 的八位输入、输出数据总线。RAM、PPU、ROM 和输入电路均与数据总线挂钩,CPU 与它们交换信息时,就是通过该数据线传递的。数据总线可工作在输入方式或输出方式,这由 CPU 的 R/W(读/写)信号决定。

• 29 脚、CLK:时钟信号输入端。由晶体振荡器产生频率为 21～27MHz 的时钟脉冲由此脚输入。

• 30 脚、TEST:测试端。生产厂家测试用,正常工作中该端接地。

• 31 脚、RDY:总线准备好信号输出端,低电平有效。

• 32 脚、IRQ:可屏蔽中断请求信号输出端,低电平有效。

• 33 脚、NMI:非屏蔽中断信号输入端,低电平有效。

• 34 脚、R/W:读、写 RAM、ROM、PPU 信息的控制信号输出端。由它决定数据总线上信息传输的方向。

• 35 脚、INP1:Ⅰ＃操纵器控制信号输出端。

• 36 脚、INP0:Ⅱ＃操纵器控制信号输出端。

INP0、INP1 信号低电平有效,为高电平时,输入电路与数据总线脱离。

• 37 脚、LOAD2:加载控制信号输出端。

• 38 脚、LOAD1:加载控制信号输出端。

• 39 脚、LOAD0:加载控制信号输出端。

当 CPU 访问输入设备(操纵器Ⅰ、Ⅱ、键盘、光电枪等)时,该端命令操作信号送入移位寄存器(4021),由 IC7 的 3 脚或 IC8 的 13 脚输出串行数据送入 CPU 的数据口 D0。

• 40 脚、V_{DD}:电源输入端(＋5V)。

1•3•3　6528 图象处理器

同 6527 中央处理器一样,6528 图象处理器也有多种型号,表 1－2 列出了不同系列机型

的 PPU 型号对照情况。

表 1－2　PPU 图象处理器型号对照表

机　型	PPU 型号	时 钟 频 率	相同功能的 PPU 型号
任天堂 天　马	6528	26.601712 或 21.47727MHz	87008、6022、KD841、 2C02、6538
小天才 胜天、科特	6528P	26.601712MHz	8008P、6541、6538P、 2D03、P03、MG－P－501

6528 与 6528P 除时钟频率略有差异外，各引脚功能完全相同。各引脚名称见图 1－5。

各引脚功能如下：

• 1 脚、R/W：来自 CPU34 脚的读、写控制信号输入端。该端为高电平时，PPU 的数据输入 CPU；为低电平时，PPU 接收来自 CPU 的图象数据。

• 2～9 脚、D0～D7：D0～D7 挂接在 CPU 的八位双向数据总线上。当 PPU 被选中(即片选信号 CE 为低电平)且 1 脚 R/W 信号为低电平时，经 CPU 控制将存入 RAM 的数据送入该数据总线，PPU 的 D0～D7 和 A0～A2 接收到 CPU 送入的图象数据后，在内部转换成地址和数据信号，由 PA8～PA13 和 AD0～AD7 输出，作为 VRAM(视频随机存储器)和 VROM(视频只读存储器)的地址信号，并与 VRAM 和 VROM 交换信息。

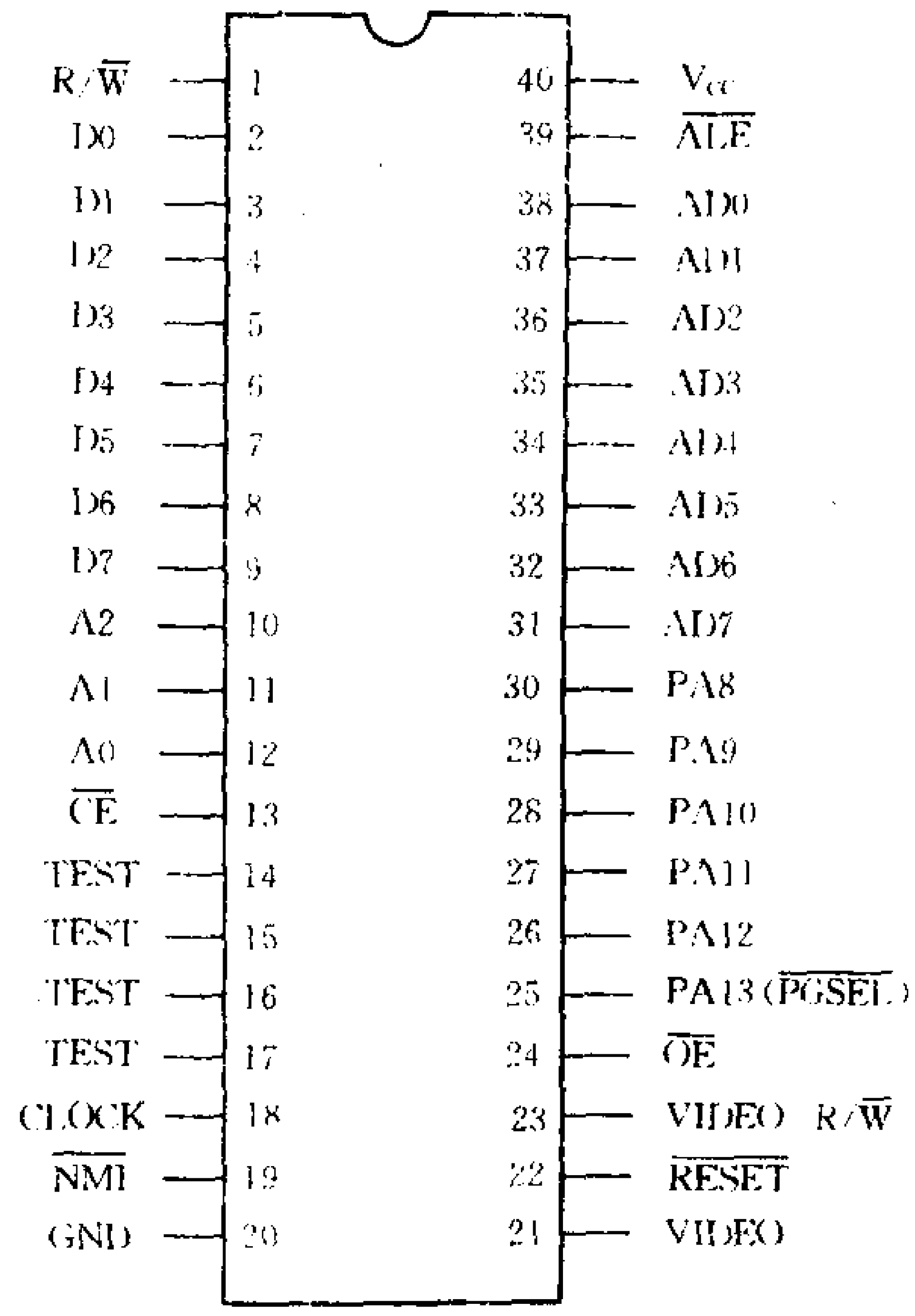


图 1－5　6528PPU 引脚排列图

• 10～12 脚、A2～A0：A0～A2 是来自 CPU 地址总线的低三位地址输入线。供 CPU 向 PPU 发出指令时寻址使用。

• 13 脚、CE：片选信号输入端。该端与译码器(139)的 1Y1(5 脚)相连，低电平有效。

• 14～17 脚、TEST：测试端，生产厂家测试用。通常该端接地。

• 18 脚、CLOCK：时钟信号输入端。时钟频率为 21. 47727MHz (或 26.601712MHz)。

• 19 脚、NMI：非屏蔽中断信号输出端。

• 20 脚、GND：地线端。

• 21 脚、VIDEO：视频信号输出端。PPU 从 VRAM 读出并行数据，经内部并/串转换后，由该端输出 PAL 制式视频信号。

• 22 脚、RESET：复位信号输入端，低电平有效。

• 23 脚、VIDEO R/W：视频信号读/写控制信号输出端。由它控制 VRAM 的 WE 和游戏卡中 VROM 的 R/W 端。

• 24 脚、OE：允许视频信号输出控制端。该端与 VRAM、VROM 的 OE 端相连，它与 VIDEO R/W 信号控制着 VRAM 的读、写和游戏卡或系统软件中 VROM 数据的读出。

• 25～30 脚、PA8～PA13：PA8～PA13 是 PPU 的地址线，它们与 AD0～AD7 一起组成 PPU 的 14 根地址线。因此，寻址能力可达 16K 字节。该地址线作为 VRAM 和 VROM 的地址

信号。因此，VROM、VRAM 可与 PPU 直接交换信息。

其中 PPU 的高位地址线 PA13 还经三态输出门电路 IC7 反相(12 脚入、11 脚出)，得到 PA13 的非信号，作为 VRAM 和 VROM 的选通信号 CE。

・31～38 脚、AD0～AD7：AD0～AD7 是 PPU 的地址、数据(双向)线。它们既作为 PPU 的输出地址线，又兼做与 VRAM 和 VROM 传输数据的数据线，这是由地址锁存器(74LS373 或 74HC373)控制转换的。PPU 的地址锁存信号使能端 ALE 连至 8D 触发器(373)的 G 端，当 AD0～AD7 作为地址输出时，ALE 为高电平，PPU 的 AD0～AD7 输出信号加至 8D 触发器的 1D～8D 端，在 ALE 变为低电平时，锁存器输出端 1Q～8Q 输出的地址信息锁存，从而构成 PPU 的 PA0～PA7 地址信号，它与 PA8～PA13 共同组成 14 根地址线，作用于 VRAM 和 VROM 的地址端。

・39 脚、ALE：地址锁存信号输出端。该端与 8D 触发器的控制端相连，低电平时锁存。

・40 脚、V_{CC}：正电源输入端(＋5V)。

1・3・4．6116 随机存储器

各种型号的任天堂系列游戏机中都使用了两片相同的 2K 字节(8×2KB)的 CMOS 静态 RAM 随机存储器。由于它们在电路中所处的位置不同、其作用和通信关系的差异，通常用 RAM 和 VRAM 来表示。其中：RAM 主要存放 CPU 的工作数据及系统堆栈；VRAM 则用于存放背景图形显示数据。6116 的引脚排列见图 1－6。

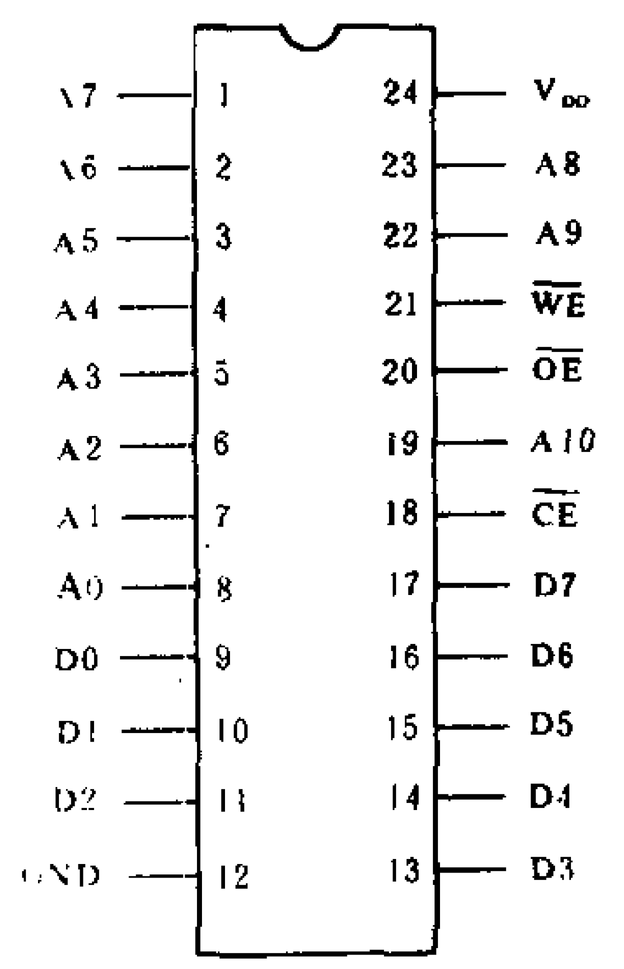


图 1－6　6116 引脚排列图

6116 各引脚的功能如下：

・1～8 脚、A7～A0：RAM 的 A0～A7 是 CPU 地址输出总线的低八位，它们与 A8～A10 组成 RAM 的 11 根地址输入线，作为查找片内数据时寻址，寻址能力为 2K 字节；VRAM 的 A0～A7 是 PPU 地址输出线的低八位，它与 PA8～PA10 组成 VRAM 的地址输入线，寻址能力为 2K 字节。而 PPU 的 14 根输出地址线是在 PPU 接收来自 CPU 的 D0～D7 位的图象数据和 A0～A2 后，由内部转换成地址信号和数据信号的，这就是 PA8～PA13 和 AD0～AD7。其中 AD0～AD7 既是数据信号线也是地址信号线，这是通过 8D 触发器 373 转换实现的。当 PPU 的 ALE 端输出触发脉冲时，8D 触发器输出 1Q～8Q，即 A0～A7 地址信号，它们与 PA8～PA13 一起组成了 14 位输出地址线 PA0～PA13，可寻址 16K 字节。

・12 脚、GND：地线端。

・9～17 脚、D0～D7：D0～D7 是 RAM 的八位数据输入、输出线，它挂接在 CPU 数据总线上。因此，RAM 与 CPU、PPU 都有通信关系。当译码器(139)的输出端 1Y0 为低电平时，即 RAM 的片选端 CE 为低电平时，RAM 被选中而开始工作，在 CPU 的控制下与 CPU、PPU 交换信息。在 VRAM 中，D0～D7 是 VRAM 的八位输入、输出线，它挂接在 PPU 的数据总线 D0～D7 上。因此，VRAM 是在 PPU 的控制下，与 PPU、VROM 单独通信的。在片选端 CE(18 脚)为低电平时，VRAM 被选中。

・18 脚、CE：片选信号输入端，低电平有效。RAM 中的 CE 端受译码器 1Y0 控制；VRAM 的 CE 是由 PPU 的输出地址线 PA13 经三态门反相后控制的。

· 19 脚、A10：A10 是 RAM、VRAM 的高位地址输入线。

· 20 脚、OE：允许 RAM 输出数据的控制信号输入端。低电平有效。在 RAM 中通常接地；在 VRAM 中由 PPU 的 OE 端控制。当 OE 端为高电平时，VRAM 的数据线脱离总线。

· 21 脚、WE：允许输入数据的控制信号输入端，低电平有效。RAM 的 WE 由 CPU 的 R/W 信号控制；VRAM 的 WE 由 PPU 的 VIDEO R/W 信号控制。

· 22～23 脚、A9～A8：A8、A9 是 RAM、VRAM 的高位地址线。

· 24 脚、V_{DD}：电源输入端（+5V）。

1·3·5 6527CPU 的工作电路

6527CPU 的工作电路见图 1－7。图中 IC2（74LS139）是低功耗肖特基 TTL2 线/4 线译码器。内部含有两个相同的译码器，现以 1～7 脚对应的 IC2b 译码器为例。其中 1G 为允许端（或称使能端，英文名称为 Enable）、1A 和 1B 为两条选择输入线、1Y0～1Y3 为四条数据输出线。

6527CPU 的基本工作原理如下：

1、CPU 的三条高位地址线 A13、A14、A15 和 RDY（READY，准备就绪）信号加到 2 线/4 线译码器 IC2 的输入端，经译码后，从 1Y0～1Y3、2Y0～2Y3 输出低电平有效的时序信号。其中 1Y0、1Y1、2Y3 分别作为 IC3 的 18 脚、IC4（6528PPU）的 13 脚及与游戏卡座的 44 脚相连 ROM 的选通信号。当 RDY 为高电平时，CPU 对外部存储器进行读写操作。当 A13～A15 均为低电平时，1Y0 为低电平，选通 RAM；当 A13 为高电平而 A14、A15 为低电平时，1Y1 为低电平，选通 PPU；当 A15 为高电平时，2Y3 为低电平，选通卡中的只读存储器 ROM。

2、CPU 的时钟信号来自晶振电路，由 CPU 的 29 脚 CLOCK 端输入，以控制 CPU 的工作节拍。

3、控制信号 INP0、INP1 控制接口电路 IC7、IC8 中三态门电路的通断，以实现与外部电路的通信。当 INP0 为低电平时，CPU 与 I ＃操纵器及扩充插座（键盘插座）通信；当 INP1 为低电平时，CPU 与 Ⅱ ＃操纵器通信。

4、CPU 的 39 脚输出 LOAD0 加载信号，LOAD0 信号接到操纵器插座和扩充插座，以对操纵器加载；LOAD1、LOAD2 接到扩充插座的第 11、第 10 脚，以作为扩充游戏功能的控制信号。如从录音磁带装回程序数据的 LOAD 信号就是由 LOAD2 发出的。

5、CPU 的 3 脚是系统复位端 RESET，当电路接通电源后，内部拉高电阻对 C1 充电。开始 3 脚为低电平，使 CPU 复位。随着充电的进行，C1 上的充电电压逐渐上升到 TTL 的开门电平时，CPU 开始工作。当按下复位开关 RESET 时，CPU 复位。CPU 复位后进入初始工作状态（即执行复位程序）。

1·3·6 6528PPU 的工作电路

PPU 的工作电路见图 1－8。图中 IC5（74LS373 或 74HC373）是低功耗肖特基 TTL8D 锁存器。

PPU 电路的工作原理如下：

1、当 PPU 对 IC6（VRAM）和卡中的 VROM 进行操作时，首先将低八位地址信号输出到 AD0～AD7，然后由 39 脚 ALE 输出一个正脉冲。在 ALE 的下降边沿作用下，把地址信号锁存在 IC5 的输出端 1Q～8Q；再从 PA8～PA12 输出高位地址信号，在 AD0～AD7 数据总线上交换信号。

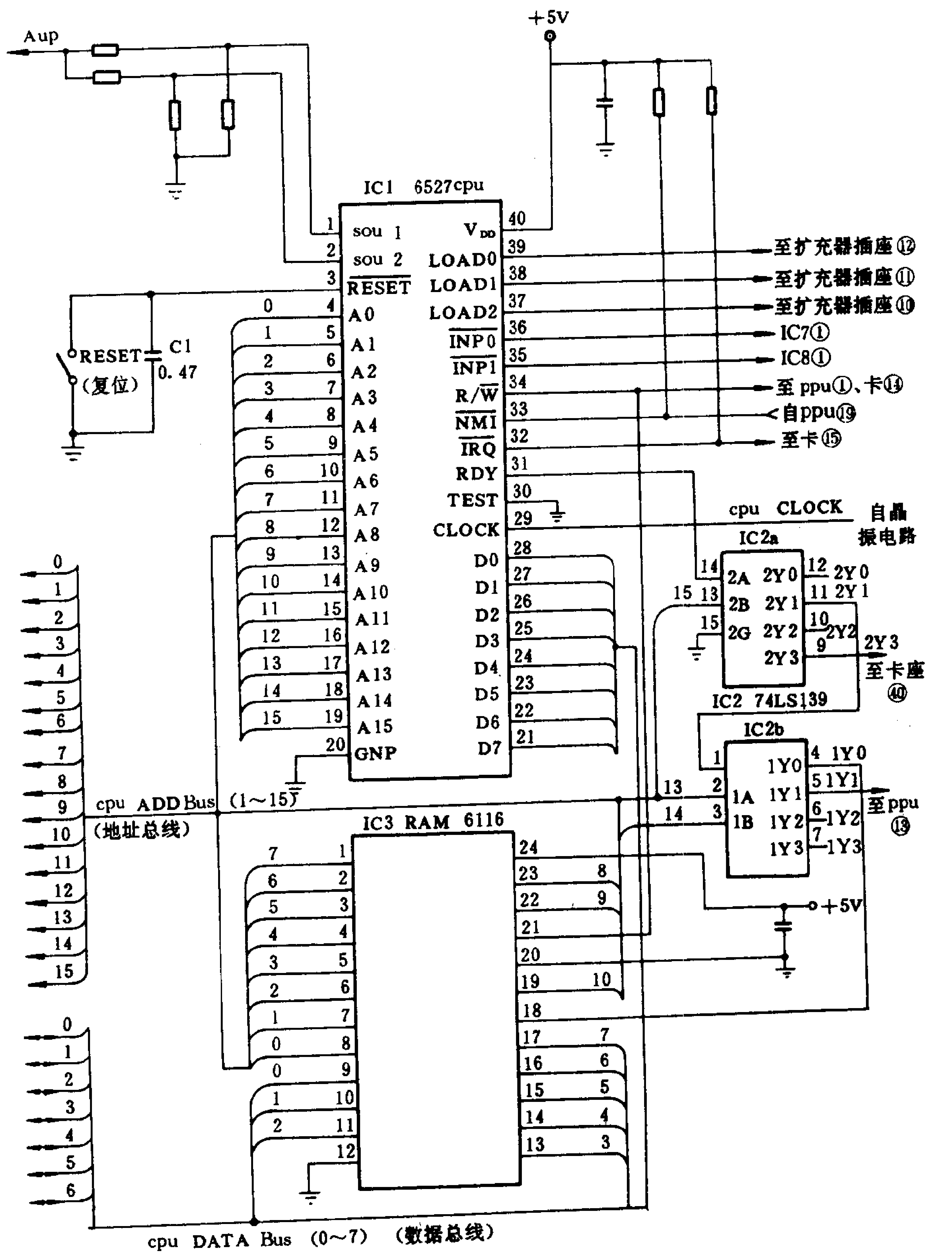


图 1—7 中央处理器(CPU)电路图

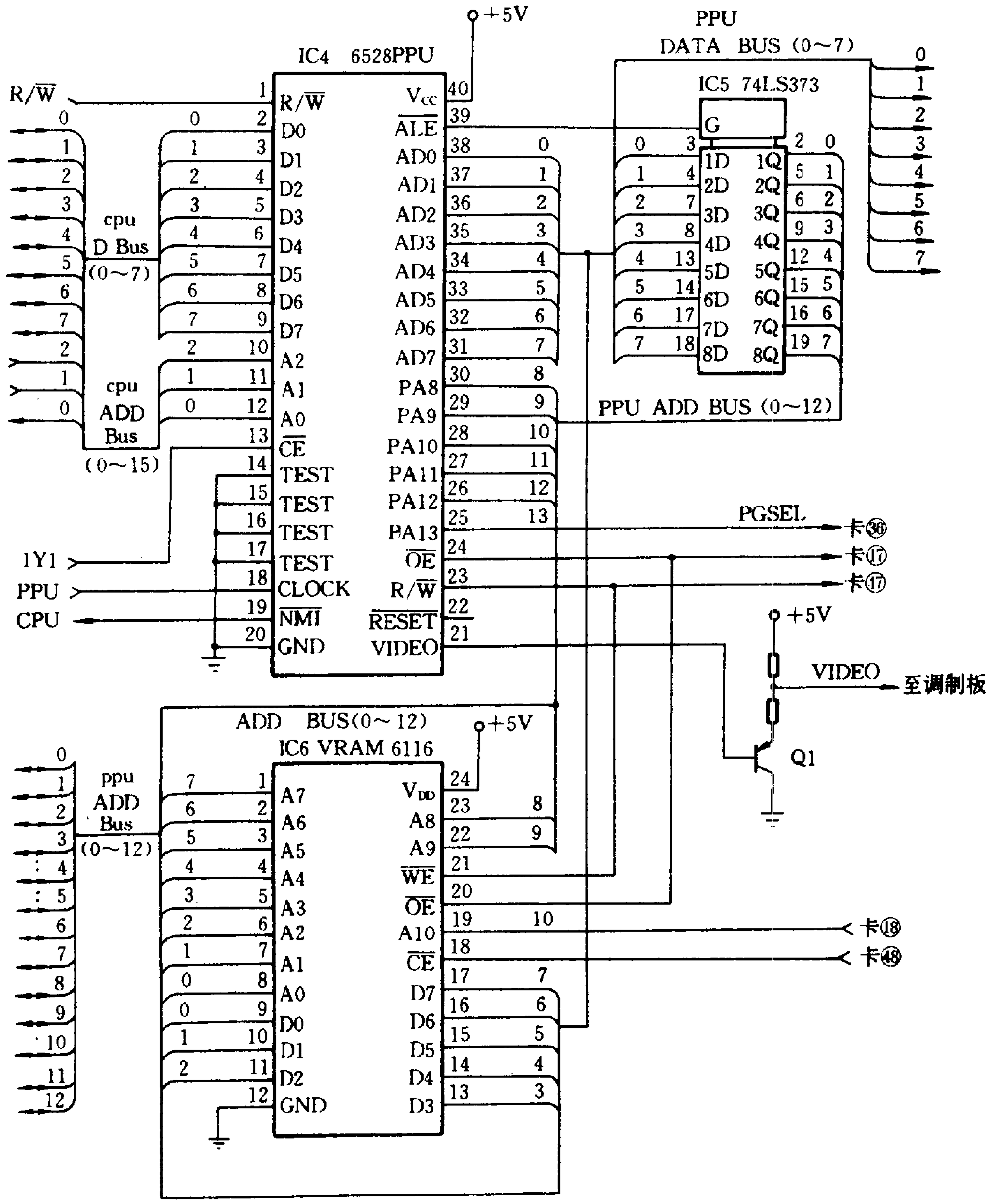


图 1-8 图象处理器(PPU)电原理图

2、PPU 的 25 脚 PA13 信号(也称 PGSEL)有两个作用:第一个作用是作为卡中 VROM 的选通信号,由 PPU 接至卡的 56 脚。当 PA13 为低电平时作为 PGSEL 信号选通 VROM,PPU 读取 VROM 中的图象数据;当 PA13 为高电平时,VROM 的数据线与 PPU 的数据线脱开,同时 PA13 经反相后变为低电平,选通 IC6(VRAM),对 VRAM 进行读写操作。由于 PA13 为高电平时才选通 VRAM,所以 VRAM 的最低地址是 2000H。PPU24 脚的 OE 信号是 IC6 的输出使能信号,同时也是卡中 VROM 的只读控制信号(由 PPU 的 24 脚到卡座的 17 脚),这就决定了 VRAM 和 VROM 只能单独与 PPU 通信。

3、当 CPU 通过 1Y1 将低电平信号加在 PPU 的 13 脚 CE 端时,PPU 被选通。CPU 通过 R/W 信号控制 PPU 的 1 脚进行读、写操作。PPU 处理接收到的信号,既产生显示所需要的行、场同步信号,又同时对 VRAM、VROM 进行读、写操作。先是按 CPU 的指令从 VROM 中读出显示数据,将其存放在 VRAM 的指定单元中,在电视扫描显示期间,PPU 把 VRAM 中的图象数据读出,进行变换、配色、重新编码,产生复合视频信号,从它的 21 脚输出,由 Q1 放大后送调制板。

1·3·7 接口电路

接口电路主要由 IC7、IC8 两片六总线驱动器 74HC368 和一片 CMOS 六反相器 CD4069 组成。其电原理图如图 1—9 所示。

74HC368 是六反相三态缓冲线驱动器,内有六个相同的高速 CMOS 反相器。每三个反相器共用一个控制端 G,分别称为 G1、G2。G1、G2 为低电平有效。当 G1=0 时,三个反相器与普通反相器一样工作;当 G1=1 时,三个反相器的输出端 Y1、Y2、Y3 便处于悬浮状态(即高阻状态)。这就是所谓的三态反相器或驱动器。其特点是除了输出为 1 和 0 两种工作状态外,还有输出端为悬浮高阻的第三态。这种三态器件主要用于驱动总线。这是因为总线上常常接有多个器件,而且有些器件并非同时工作,通常在同一时刻只允许一个器件驱动总线,此时不驱动总线的器件应"脱离"总线,而上述第三态便可实现自动脱离总线的目的。

工作时,从操纵器、光电枪、键盘等外设发出的外部控制信号都将通过接口电路挂在 CPU 数据总线上。操纵器 Ⅰ、Ⅱ 分别通过游戏机电路板上的插座 CZ1、CZ2 与接口电路相连,其它控制附件如光电枪、键盘等均通过扩充插座 CZ3 与接口电路连接(集合式结构已在内部连接)。

进行游戏时,CPU 的 INP0 为低电平,选通了 IC7 的 F1~F4,CPU 的 LOAD0 信号分别通过 CZ1、CZ2 的 3 脚对操纵器 Ⅰ、Ⅱ 加载;CPU 的 RDY 信号经 IC7—F3 反相后送到操纵器的 2 脚,于是操纵器 Ⅰ 发出的串行控制信号(数据 DATA)经 IC7—F1 反相后驱动 CPU 的数据线 D0,从而控制游戏的进行。此时,CPU 的 INP1 为高电平,IC8 的 F1~F6 悬浮不通,Ⅱ # 操纵器没有 CLOCK 信号,其 DATA 信号与 D0 数据线脱离,因而 Ⅱ # 操纵器无效。当第一游戏者游戏结束后,CPU 把 INP1 置低电平,INP0 置高电平,使 Ⅰ # 操纵器的 DATA 信号脱离 D0 数据线,而 Ⅱ # 操纵器的 DATA 信号挂通 D0 线,这样 Ⅱ # 操纵器有效,Ⅰ # 操纵器无效。这就是两人轮流游戏的情况。在双打游戏中,两个操纵器将同时控制同一画面中的不同角色。这时,CPU 将分时轮流访问两个操纵器,即使 INP0、INP1 轮流为低电平,两人就可同时游戏。显然,两个操纵器的分时传输也是由接口电路实现的。键盘、光电枪、录音机的数据信号也通过接口电路挂接在 D0~D4 数据线上。IC7 的 15 脚 G2 接地,故 F4~F6 作为一般反相驱动器使用。

从 CPU 电路输出的音频信号 AUP 通过 1μF 的电容器加在 IC7—F6 的输入端,通过 IC7—F6 反相放大后送到调制板。

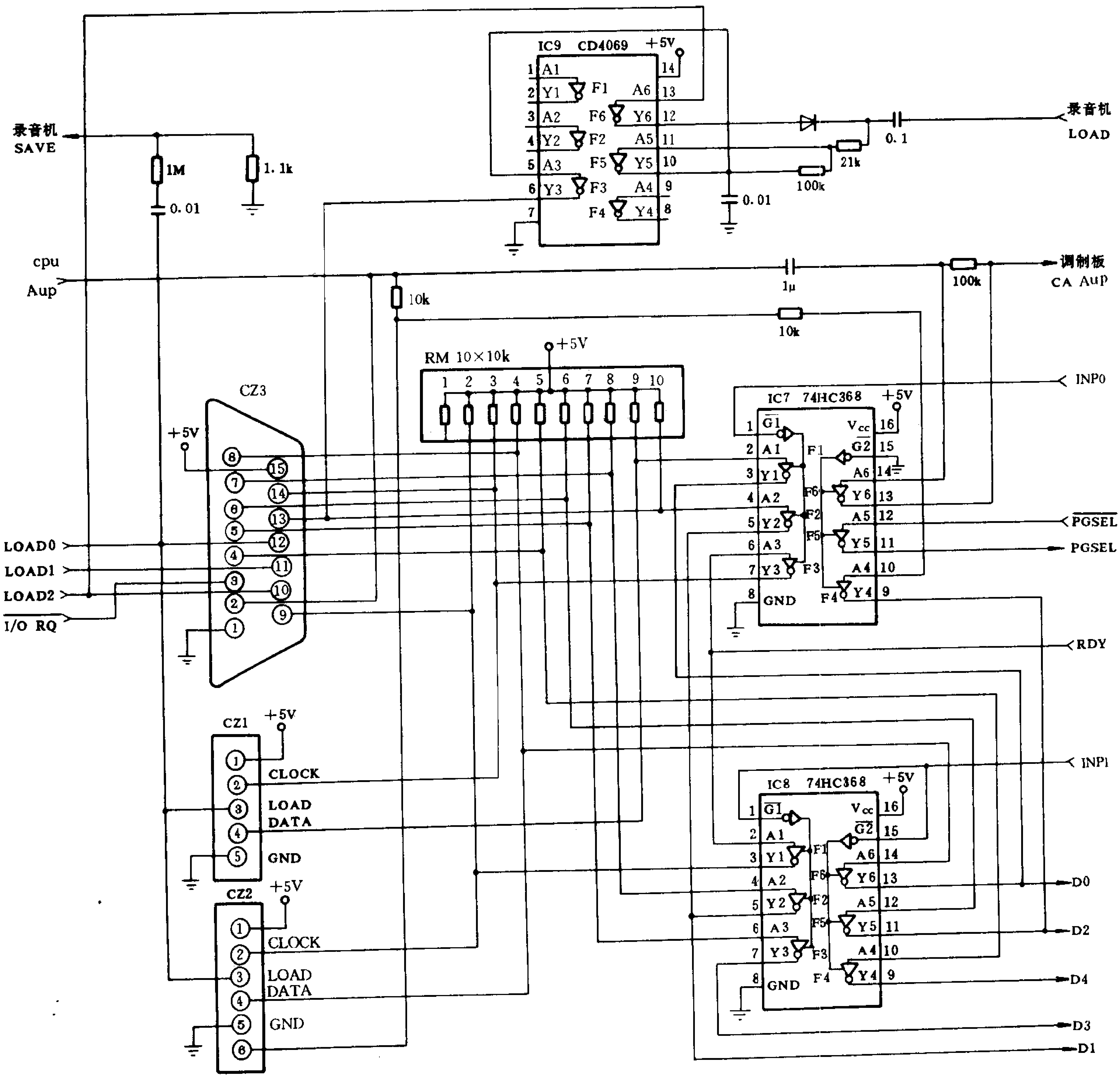

IC9 CD4069
+5V
录音机
SAVE
录音机
LOAD
1M
1.1k
0.01
0.1
21k
100k
0.01
cpu
Aup
1μ
100k
调制板
CA Aup
10k
10k
RM 10×10k
+5V
CZ3
INP0
IC7 74HC368
PGSEL
PGSEL
LOAD0
LOAD1
LOAD2
I/O RQ
RDY
CZ1
CLOCK
LOAD
DATA
GND
INP1
IC8 74HC368
CZ2
CLOCK
LOAD
DATA
GND
D0
D2
D4
D3
D1

图 1—9 接口电路原理图

从录音机 EAR 端口输出的数据信号经 IC9—F5、IC9—F3、IC7—F2 放大后送入 D1 数据线。

RM 是集成电阻排，它是把若干只电阻封装在一起的电阻块。这样缩小了体积、提高了可靠性，使装配更加方便。

15 芯插座的引脚定义如表 1－3。

表 1－3　15 芯插座引脚定义说明

引脚编号	引　脚　定　义
1	电源地线
2	提供音频信号输入接口
3	与 CPU 的 RQ 相连
4	与 INP1 合成后送数据线 D4，供 CPU 读取。
5	与 INP1 合成后送数据线 D3，供 CPU 读取。
6	与 INP1 合成后送数据线 D2，供 CPU 读取。
7	与 INP1 合成后送数据线 D1，供 CPU 读取。
8	与 INP1 合成后送数据线 D0，供 CPU 读取。
9	与 INP1 合成信号，提供 I # 操纵器的时钟信号 CLOCK。
10	与 CPU 的 LOAD2 相连，控制外设。
11	与 CPU 的 LOAD1 相连，控制外设。
12	与 CPU 的 LOAD0 相连，控制外设。
13	与 INP0 合成信号送至 D1，由 CPU 读取。
14	与 INP0 合成信号，提供 I # 操纵器的时钟信号 CLOCK。
15	＋5V 电源

1·3·8　时钟电路

电脑游戏机的时钟电路安装在游戏机的主控印刷电路板上。由于游戏机型号的差异，时钟电路也略有不同，下面列出典型的三种时钟电路。见图 1－10、图 1－11、图 1－12。

从图 1－10、1－11 可见，两图的差异仅仅是图 1－10 中的振荡电容 C 并联了一个可变电容 C′，以便于校准晶体振荡频率。早期的任天堂游戏机多为这种电路。图中 Q1、TX、C1、C2 构成时钟发生器，产生的高频脉冲用作 CPU 和 PPU 的时钟。时钟脉冲一路经 C4 到 PPU 的第 18 脚；另一路经射极跟随器 Q2 、C5 到 CPU 的 29 脚。高频脉冲的频率由晶体 TX 决定，常见的有：21.47727MHz、21.251465MHz 以及 26.601712MHz 三种，这是根据不同的机型，使用不同的 CPU、PPU 而定的。图 1－12 为双时钟电路，它适用于 CPU 使用 21.47727MHz 、PPU 使用 26.601712MHz 的电路。

图 1－13 列出了游戏机主板上三种不同时钟频率与 CPU、PPU 的配合关系。

图 1－12 的三种配合，在实际应用中的对比情况如下：

图 1－13(a)的配合是任天堂原装机的方案。这种原装机电路板质量较高，工作稳定、兼容

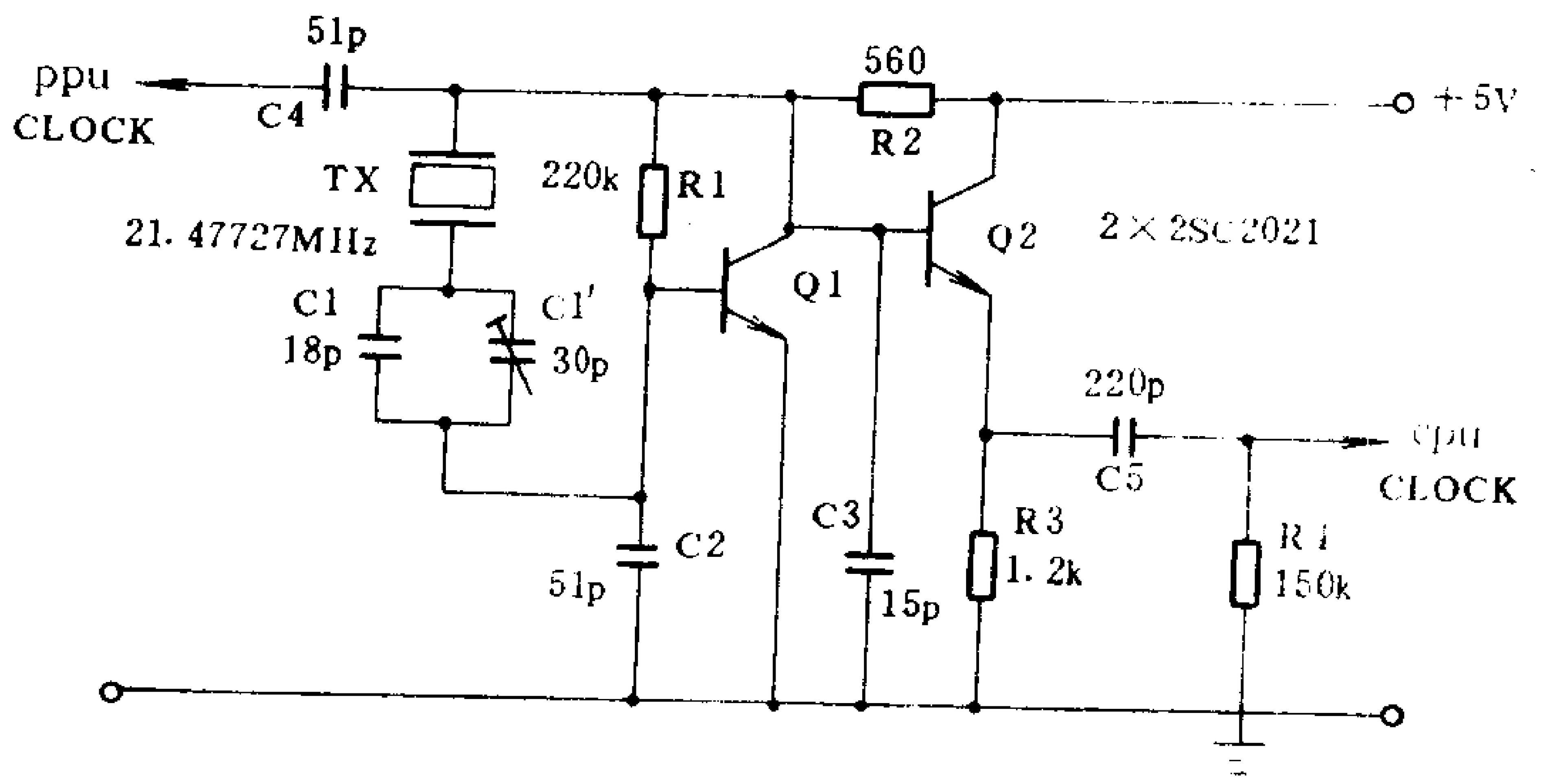


图 1－10 时钟电路之一

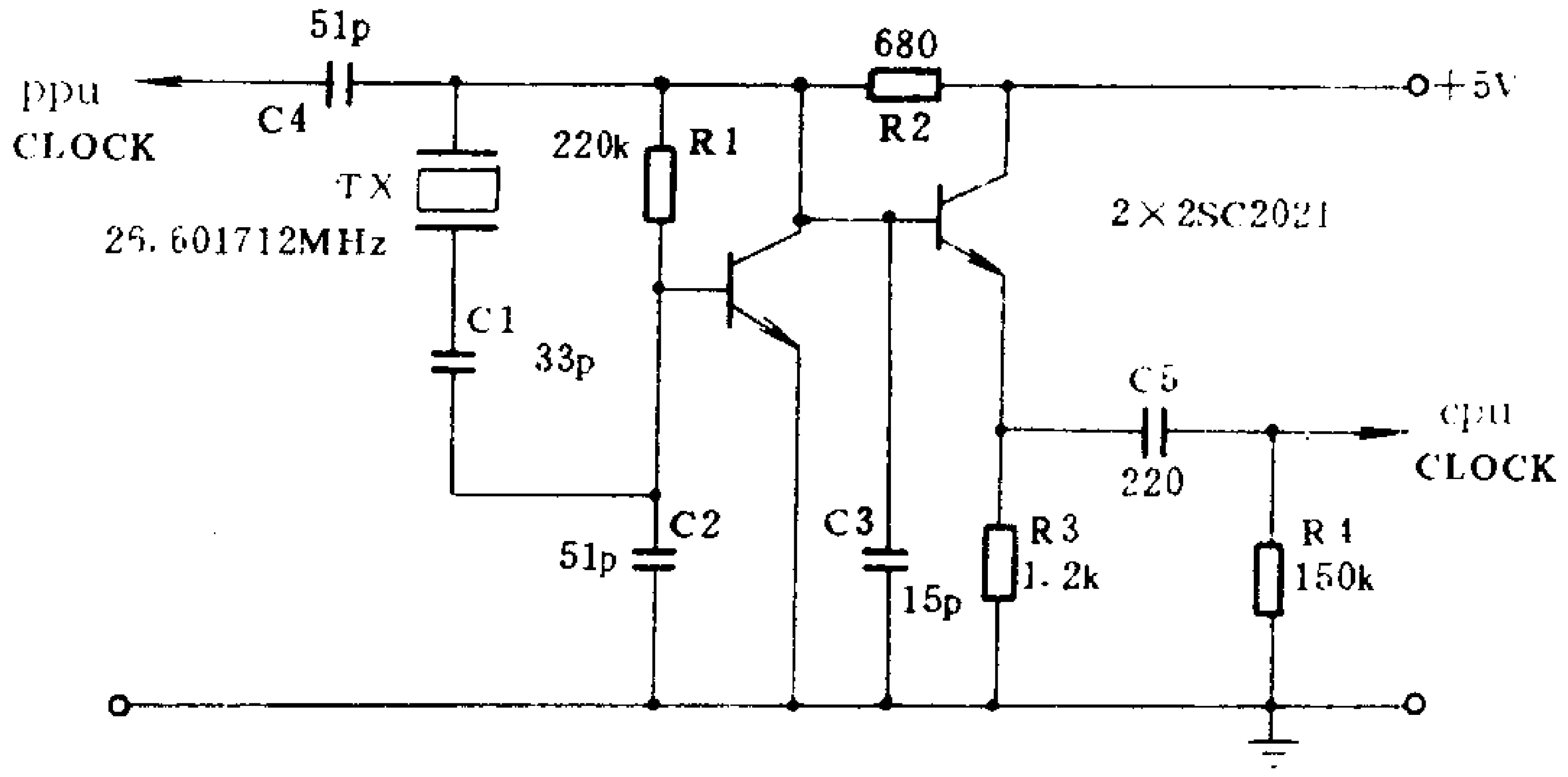


图 1－11 时钟电路之二

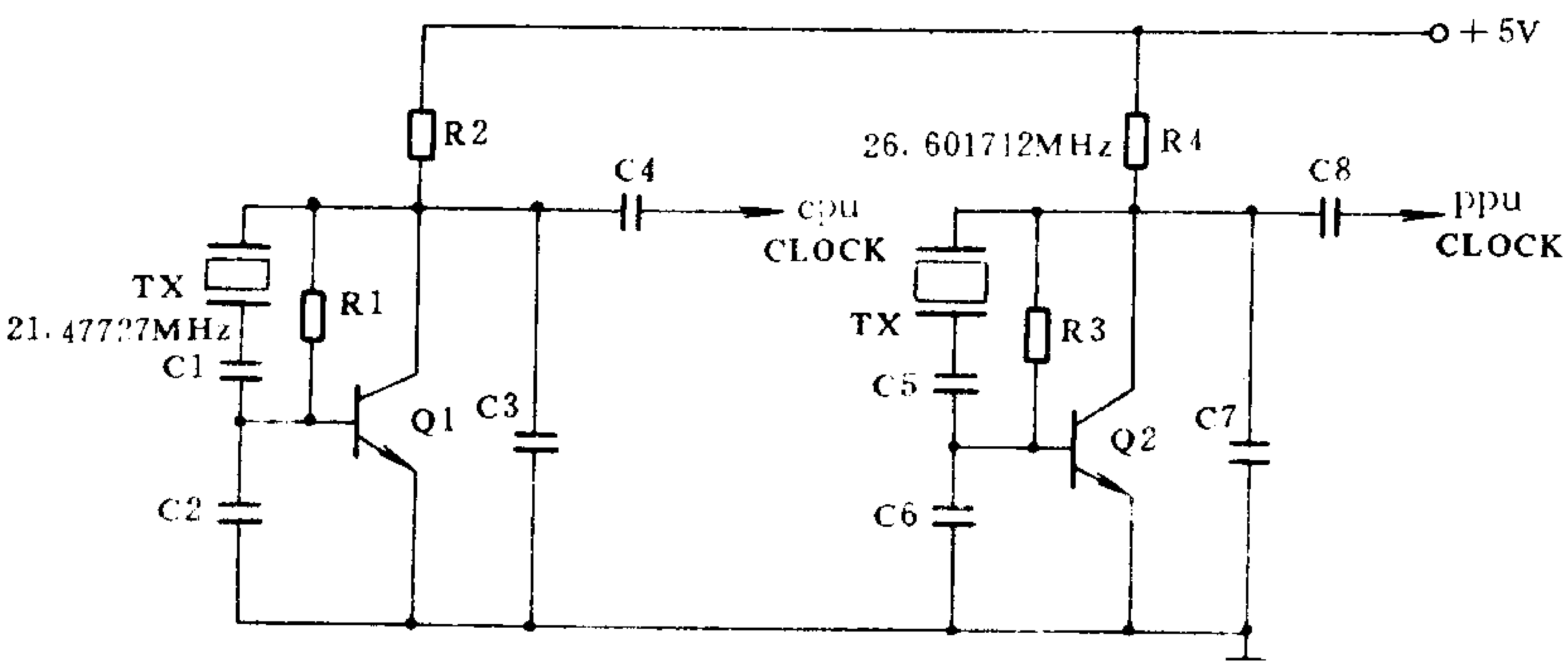


图 1－12 时钟电路之三

性好。缺点是输出的视频信号是 NTSC 制式，在我国使用时要加制式转换电路。

图 1－13(b)是一种过渡性方案。它虽然能输出 PAL－D 制式的视频信号，但由于 CPU、PPU 的时钟频率相互独立，故极易相互干扰，致使图象质量不太理想。

图 1－13(c)是一种最佳的方案。现在流行的仿任天堂红白机、小天才、小灵通等机种，大多采用这种方案。这种方案的缺点在于 CPU、PPU 大多是港、台等地根据 PAL－D 制式的要求而重新设计制造的，因而游戏卡的兼容性稍差些。

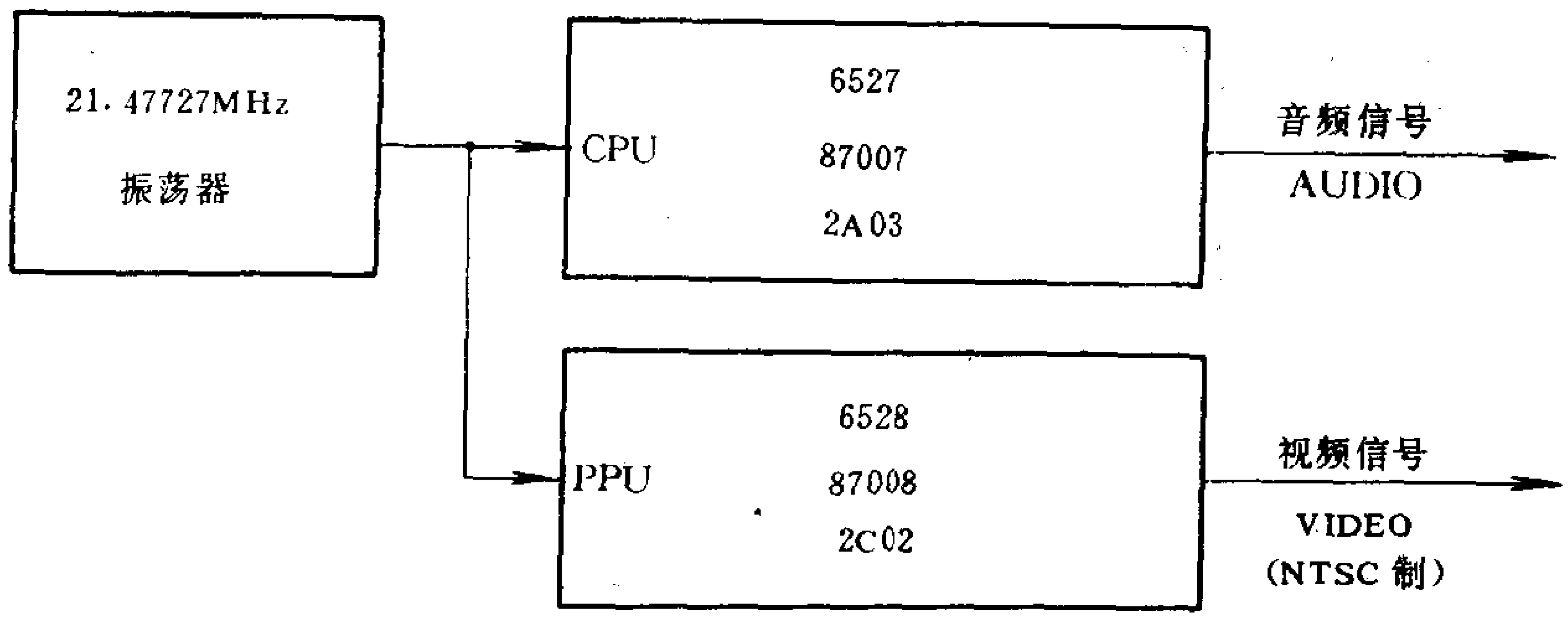


图 1－13(a) 时钟频率与 CPU、PPU 配合图之一

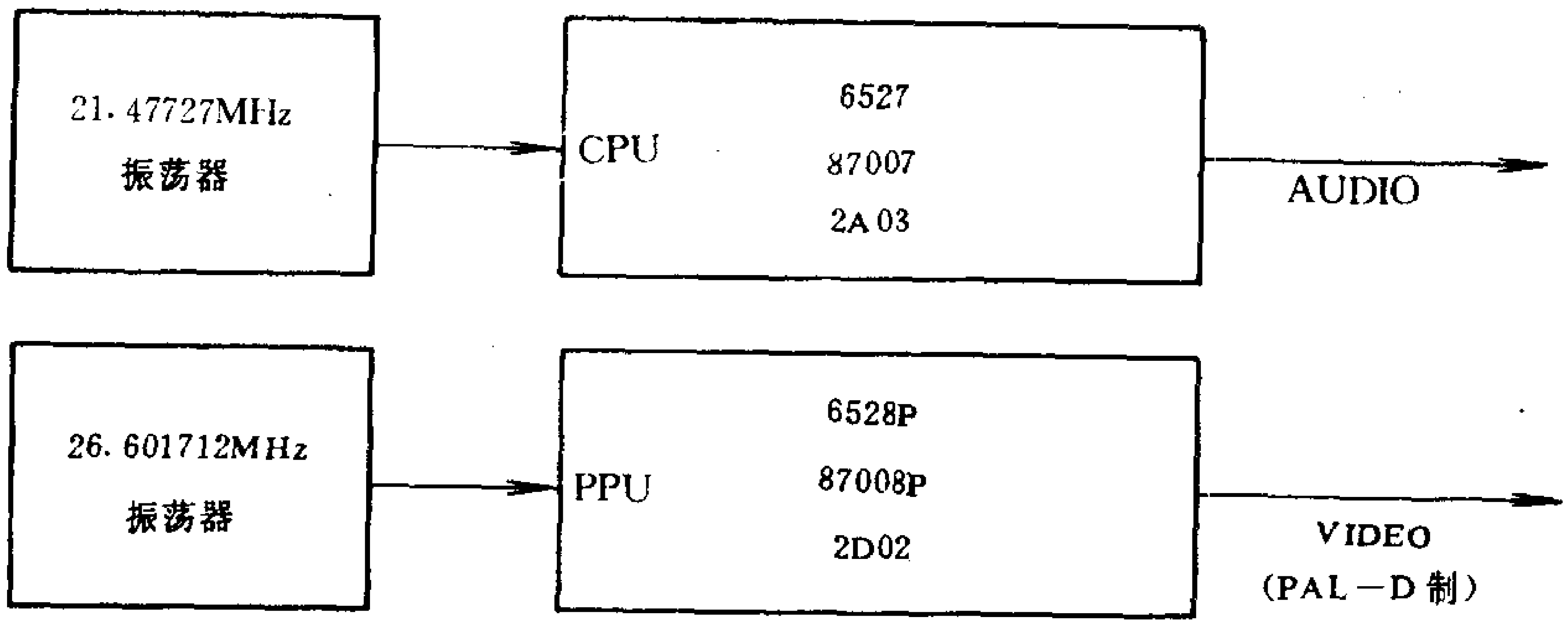


图 1－13(b) 时钟频率与 CPU、PPU 配合图之二

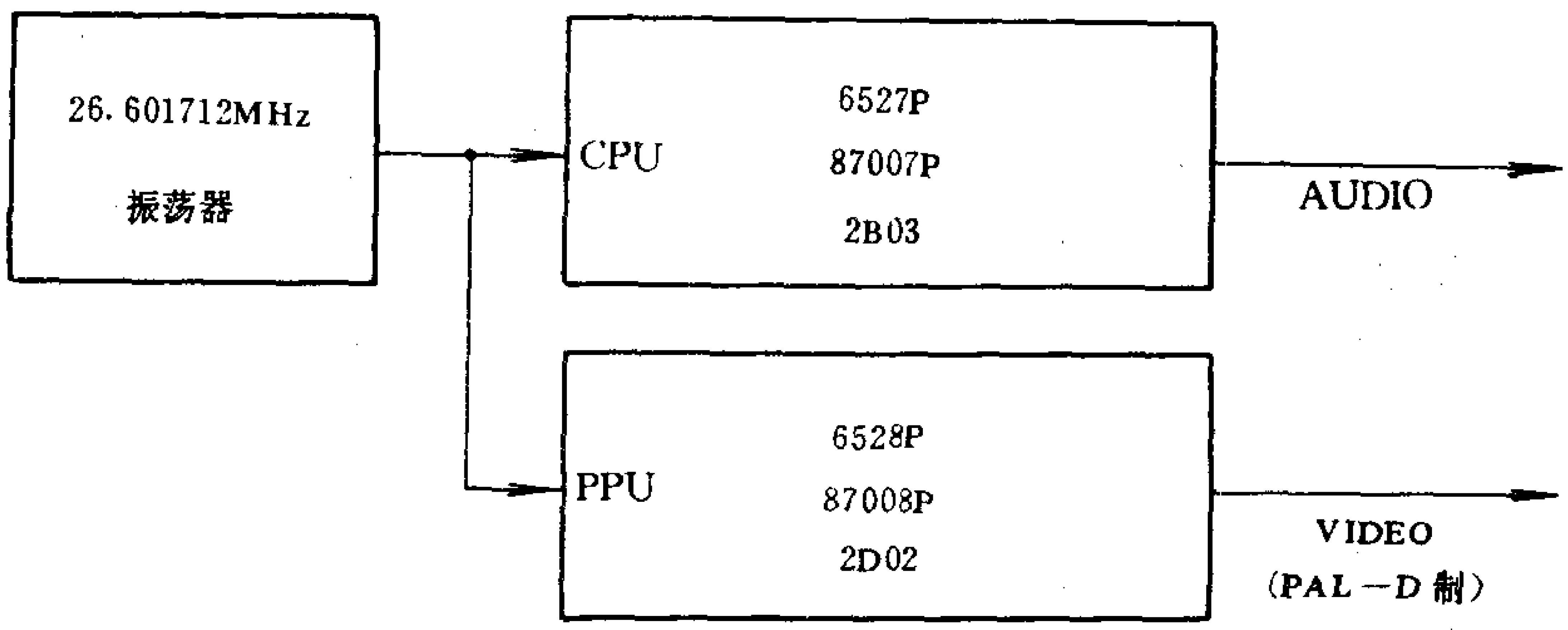


图 1－13(c) 时钟频率与 CPU、PPU 配合图之三

1·3·9 盒卡简介

我们知道，任天堂游戏是在游戏卡的支持下进行的，换用不同的游戏卡可进行不同的电视游戏；而电脑游戏机的工作是在学习卡的支持下进行的(对分立结构而言、下同)，换用不同的学习卡，可进入不同的工作系统。可见，游戏卡、学习卡也是电脑游戏机的重要组成部分。实际上，游戏卡、学习卡具有相同的硬件结构，差别仅是存储器中固化的软件不同和存储器容量的差异。因此，我们把游戏卡、学习卡统称为盒卡。

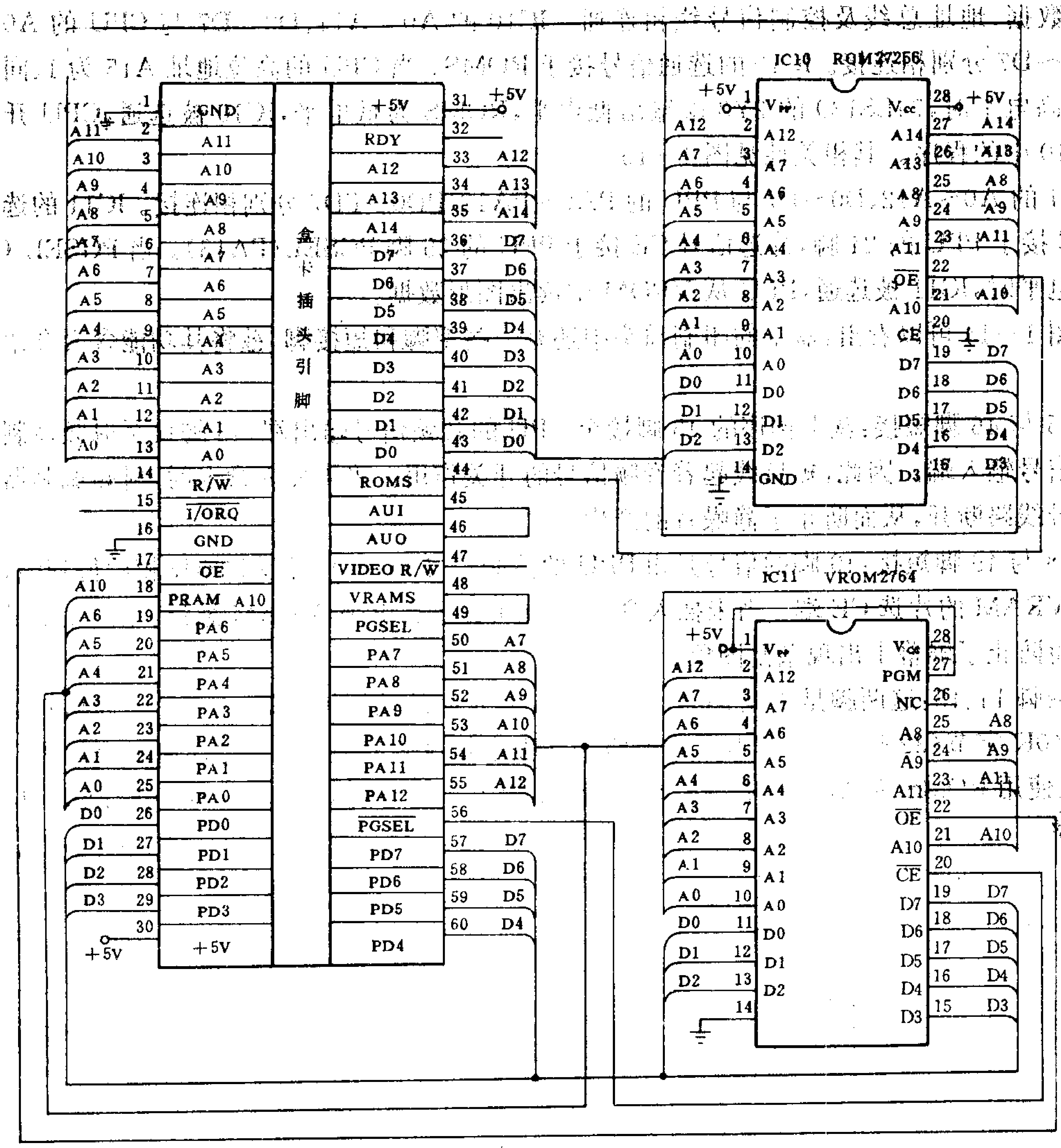


图 1－14　盒卡逻辑原理图

图 1－14 是盒卡的基本逻辑原理图。图中：IC10 为程序存储器，内存游戏管理程序或学习工作系统软件；IC11 为数据存储器，内存各种图形的数据（字模）。IC10、IC11 的各引脚通过盒卡印刷电路板上的 60 脚印刷插头与主机相连。当盒卡插入主机插座时，IC10、IC11 便与 CPU、PPU 的数据、地址总线及控制信号线相连通。IC10 的 A0～A14、D0～D7 与 CPU 的 A0～A14、D0～D7 分别相连接。IC10 的选通信号接于 ROMS。当 CPU 的高位地址 A15 为 1、同时 RDY 为高电平时，74LS139 的 2Y3 端输出低电平，ROMS 为低电平，IC10 被选通，CPU 开始执行 IC10 中的程序。其相关性见图 1－15。

IC11 的 A0～A12、D0～D7 与 PPU 的 PA0～PA12、PD0～PD7 分别相连接。IC11 的选通信号 OE 接于 PPU 的 24 脚，片选信号 CE 接于 PPU 的 25 脚 PGSEL(PA13)。当 PGSEL、OE 都为低电平时，IC11 被选通，PPU 从 VROM 中读出图形数据。

从图 1－14 可以看出，盒卡的印刷插头中还有一些空脚和短接脚，兹将其功能含义介绍如下：

1、45 与 46 脚短接：盒卡插座的 45 脚接至 CPU 的音频信号输出端，46 脚接至射频调制器的音频信号输入端。因此，短接线起着音频信号的开关作用。当未插入盒卡时，射频调制器的音频信号线路断开，从而防止了静噪音的产生。

2、48 与 49 脚短接：49 脚的信号是由 PPU 的控制信号 PGSEL 经反相器反相后得到的，48 脚接至 VRAM 的片选 CE 端。当未插入盒卡时，因 CE 呈高电平，VRAM 悬浮，无图象信号输出。从而防止了屏幕上出现杂乱图象。

3、空脚 14、15：这两脚是扩充 ROM 和 VROM 容量时的控制信号脚。图 1－14 是直接寻址空间为 40K 字节（32＋8K）的基本逻辑电路。当程序超过 40K 时，就要对存储器容量进行扩充。一般使用 ROMS、R/$\overline{W}$ 信号扩充存储器容量。图 1－16 是把 VROM 的容量扩充到 32K 的外围电路。

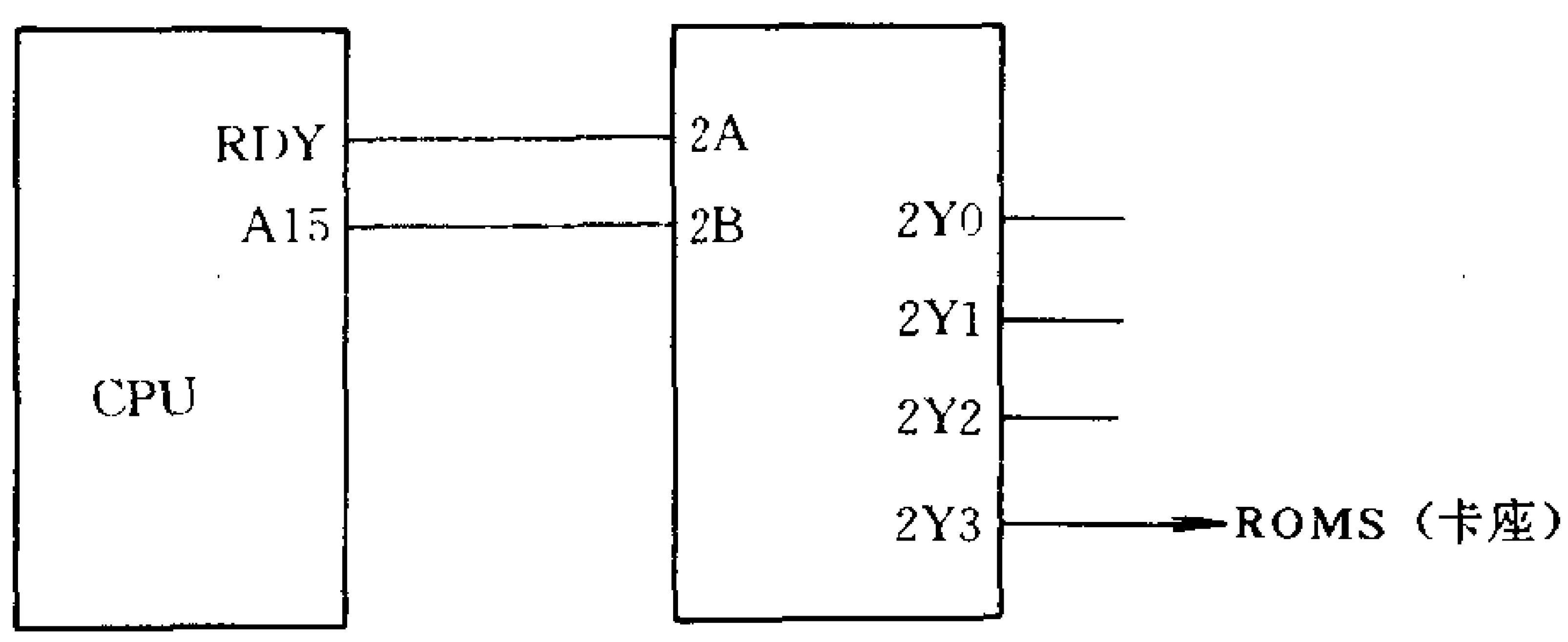


图 1－15　CPU 与盒卡的相关图

图 1－16 中 74LS161A 是高速二进制可预置同步计数器，设定为预置工作状态。在 CE 和 R/$\overline{W}$ 信号的控制下，由 PPU 数据线 D0、D1 传送的信号被预置入 74LS161 的 A、B 中，由 QA、QB 端控制 VROM 的高位地址 A13、A14 的状态，从而把 VROM 的容量扩充到原来的四倍。

图 1－17 是把程序存储器 ROM 的容量扩充到 554K 的外围电路。

图 1－17 中把 ROM 看作一个虚拟存储器，它由 17 个存储体组成，其中包括一个基本存储体和 16 个扩充存储体，各存储体采取段切换方式进行选择。基本存储体由 CE0 和 OE0 选通，扩充存储体分别由 CE1～CE8、OE1～OE8 和 CK 选通。由于每个存储体的容量为 32K 字节，所以 ROM 的最大存储容量被扩充到(1＋8×2)×32＝544K。

4、空脚 47：47 脚接于 PPU 的图象读、写信号端 VIDEO R/W。有的大容量节目卡，除对 ROM 和 VROM 的容量进行扩充外，还装有 1～8K 的 RAM 供用户设计背景图形使用。此时，利用 VIDEO R/W 信号对 RAM 进行读、写操作。

关于电脑游戏机硬件系统的其它部分，如光电枪、操纵器、键盘、射频调制器、稳压电源以及软驱、打印机接口等电路，由于结构比较简单，故不多赘述。

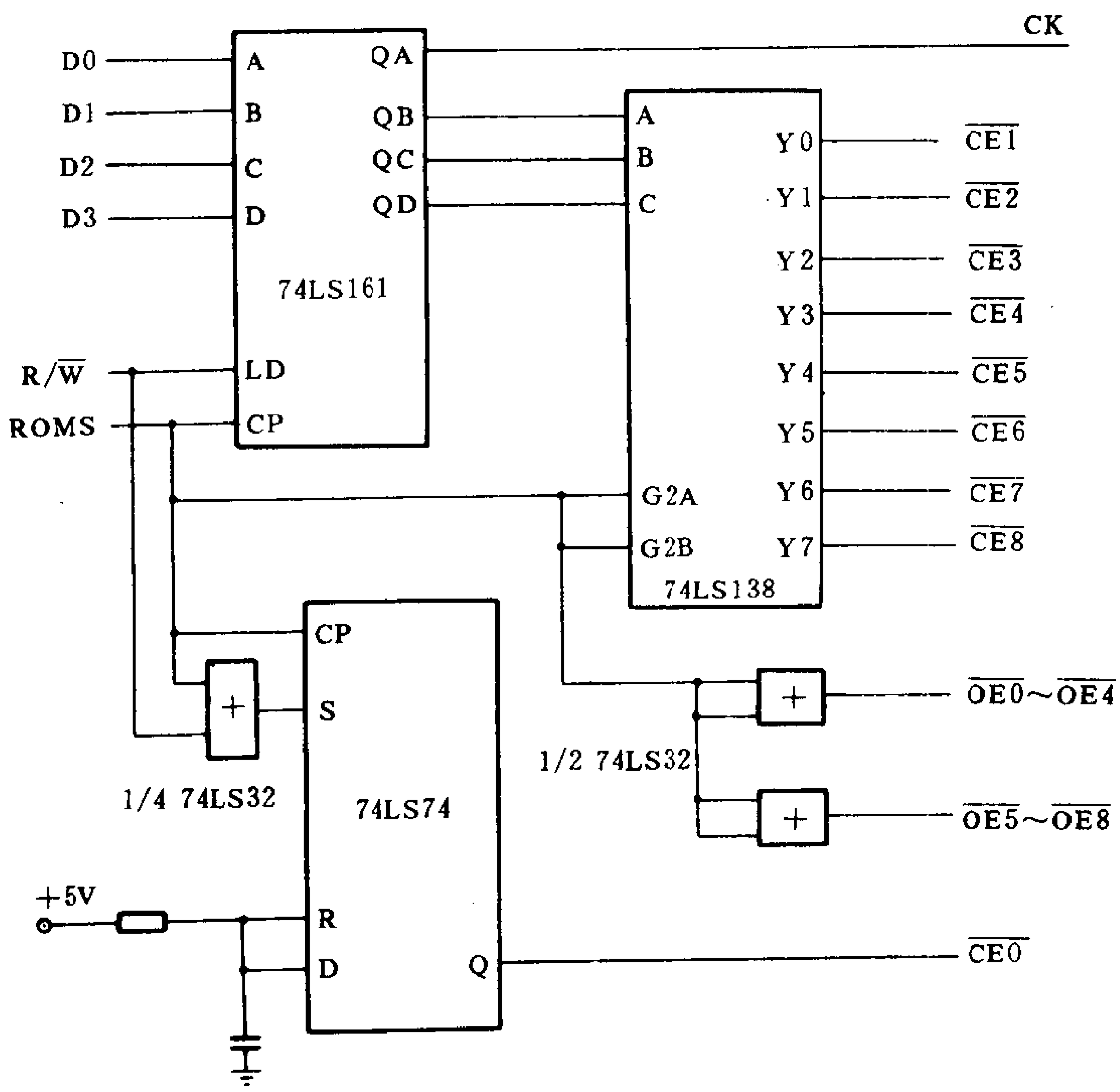


图 1－17 ROM 扩充容量的外围电路

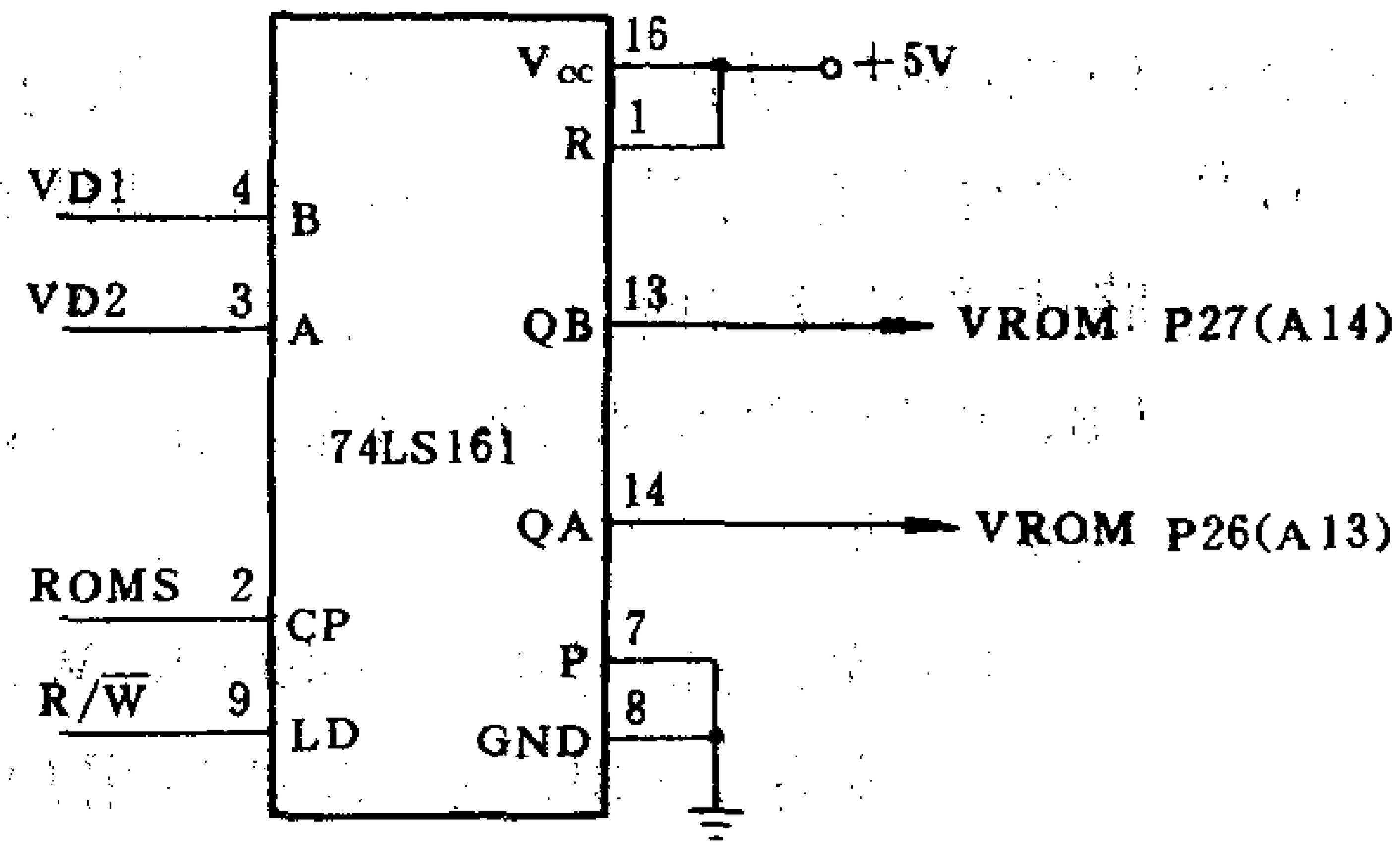


图 1－16 VROM 扩充容量的外围电路

1·4 电脑游戏机的软件系统

由图 1－1 可见，电脑游戏机的软件系统共分两大类：一类是系统软件；一类是应用软件。作为电脑游戏机工作系统的入门篇，本书重点介绍系统软件。应用软件则另文阐述。

1·4·1 6527CPU 工作系统

通过硬件系统的简单分析可以看出，与中华学习机相比，电脑游戏机比它多出了一块图象处理器——6528PPU。PPU 不仅具有自己的时钟电路，而且还具有与 CPU 相同的访问 ROM、RAM 的能力。在以后的分析中，我们将会看到：正因为引进了 PPU，任天堂游戏机才能够展现出声、像同步、各自独立工作的奇妙效果，使一个 CPU 发挥了两个 CPU 的效能。这一点也正是任天堂游戏机设计者独具匠心的高明之处。

但是，无论 PPU 的功能如何强大，它仍是在 CPU 指挥协调下工作的一部分，仍然从属于 6527CPU。因此，我们所说的 6527CPU 工作系统也包括 PPU 在内。

在 1·3·1 节的讨论中，我们曾谈到 6527CPU 不仅是一个中央处理器，而且还是一个八位单片计算机，它的内部有一套独特的工作系统支持着它的工作，从而使电脑游戏机工作过程中的许多操作，不必用户编程控制，只须给它一个简单的指令或触发某一个软开关就可以由 CPU 自动完成。如仅对 $ 2005 单元置入不同的数值，就可以控制画面横向卷动或纵向滚动。可见，与 6502CPU 相比，6527CPU 不仅图形处理能力强，而且操作控制简单、容易。它有一套独具一格的工作系统。

6527CPU 的工作系统固化在 CPU 内部，一般很难读出研究，又加之有关 6527CPU 的资

料短缺，因此人们对 6527CPU 的了解目前尚处于试探、摸索、研究、逐步认识阶段。业余研究的唯一途径就是通过阅读电脑游戏机的各种工作系统软件，如：人机对话工作系统、F BASIC 工作系统以及任天堂游戏源程序等，从中归纳、汇集、总结出它的一般规律后，再进行上机验证，最终确定功能。这显然是一项复杂、艰巨的繁琐工作。因此，本书的编写宗旨并不在于全面介绍 6527CPU 的工作系统，而在给出研究它的一般方法，给大家提供一条研究 6527CPU 的捷径。所以本书仅以特点的方式介绍 6527CPU 的工作系统。书中所述内容则是积累了近年来国内同仁的研究成果和笔者的研究心得，有些问题的表述、解释可能不甚贴切，还有待广大同仁验证、斧正。

1·4·2　人机对话工作系统

人机对话工作系统是学习卡中的主要系统，当前流行的所有电脑游戏机几乎全部保留了这一工作系统。该系统集中演示了任天堂游戏中的基本操作控制过程。如 32 列×30 行背景画面的绘制；背景配色的设定与改变；卡通动画的运动控制；发声过程的处理；声像同步技巧；效果音响的实现以及背景 0、1 页的转换等等。因此，人机对话工作系统软件将是本书讨论的重点。

1·4·3　F BASIC 工作系统

F BASIC 工作系统是电脑游戏机的重要工作系统。F BASIC 编程、BG GRAPHIC 键控绘图都在该工作系统下进行。由于目前流行的键盘几乎都未配备监控系统和汇编系统，因此编写程序仍离不开 BASIC 语言。我们知道，用 F BASIC 语言编写的程序运行速度慢、内存开销大、工作效率低，而电脑游戏机配备给用户的内存又太少（一般 4～8K），因此迫切需要占用内存少、运行速度快的机器语言编程。在以后的讨论中将会看到，F BASIC 指令中的 POKE 指令和 CALL 指令，为我们使用机器语言编程和运行机器语言程序提供了方便。因此，研究 F BASIC 工作系统、掌握各条指令的执行过程和操作实质，既知其然又知其所以然，就可在以后的编程中灵活调用各系统的独立子程序，达到事一分而功十倍的目的。

由于 BS. 2A 版本是所有键盘必配的工作系统，所以本书以 F BASIC 的 BS. 2A 版本为基础进行讨论。但总结出的基本规律却适用于一切 6527CPU 的工作系统，因而具有广泛的普遍性。

第二章　6527CPU 工作系统的特点

2・1　6527CPU 的基本特征

2・1・1　6527CPU 的指令系统

6527CPU 属于 65 系列微处理器，它具有 65 系列 CPU 的基本特征。它的指令系统与 6502CPU 的指令系统完全兼容。故凡学习过苹果机、中华学习机机器语言的朋友，阅读 6527 机器语言程序则是驾轻就熟。(6527CPU 的指令表见附录)

2・1・2　6527CPU 的内部寄存器

同 6502CPU 一样，6527 内部也有六个可供使用的寄存器(A、X、Y、S、P、PC)，其中除程序计数器 PC 是十六位之外，其它寄存器均为八位。

为便于掌握使用，特对各寄存器的性能、作用简介如下：

1、累加寄存器 A(Accumulator)

它是一个八位寄存器，一般称为累加器或 A 寄存器。它同算术逻辑运算单元 ALU 一起完成各种算术逻辑运算。在通常的工作中，它既为 ALU 提供原始操作数据又担任存放 ALU 的运算结果的任务。因此，在各寄存器中它是最活跃、使用最频繁的寄存器。

2、变址寄存器 X、Y(Index Register)

变址寄存器共有两个，均为八位寄存器，分别用大写字母 X、Y 表示。它们主要用在变址寻址方式中存放地址偏移量(变址器)，也常被当做计数器使用，还可作为通用寄存器用于暂存数据。

3、堆栈指针寄存器 S(Stack Pointer)

它是专用于指示系统堆栈栈顶位置的八位寄存器。由于 6527CPU 规定系统堆栈区设置在第一页，所以堆栈指针 S 有八位已能满足使用。工作中它指向堆栈地址的低八位，在有数据进栈时，S 内容自动减一；数据出栈时，S 内容自动加一。

4、程序计数器 PC(Program Counter)

PC 是 6527CPU 中唯一的 16 位寄存器。它主要用于存放下一条指令的首字节地址。当程序顺序执行时，每取出一个指令字节后，PC 自动加一，为取下一个指令字节作好准备；当程序需要转移到新的地址时，则 PC 中将放进要转移的目标地址。

5、标志寄存器 P(Processor Status)

标志寄存器 P 也称状态寄存器，用大写字母 P 表示。它也是一个八位寄存器，工作中只使用了七位，D5 位未用。这七个标志位的作用如下：

D7	D6	D5	D4	D3	D2	D1	D0
N	V		B	D	I	Z	C

·C——进位标志(Carry)。它标志着指令执行后最高位的进位状态。进行加法运算时,若最高位有进位则 C 位置 1、否则置 0;进行减法运算时,若最高位有借位则 C 位置 0、否则置 1;进行逻辑运算时 C 位置 0。

·Z——零标志(Zero)。它表示本次操作后执行结果的情况。若结果为 0 则 Z 位置 1,否则置 0。

·I——中断禁止(屏蔽)标志(Interrupt Disable)。它表明是否允许中断。允许中断时 I 位为 0,禁止中断时 I 位为 1。由于 6527CPU 的工作中一般仅使用非屏蔽中断,所以该位一般不用。

·D——十进制运算标志(Decimal Mode)。它表示 ALU 进行的运算是十进制运算还是二进制运算。十进制运算时 D 位置 1,否则 D 位置 0。D 位的状态只对后续的加法或减法指令有作用。它只是在运算前对运算数据做出规定而不表示运算的结果。

·B——软件中断指令标志(Break Flag)。它表示软件中断状态。当执行 BRK 指令后 B 位置 1,否则 B 位置 0。

·V——溢出标志(Overlow)。它表明本次运算结果的溢出状态。若有溢出则 V 位置 1,否则 V 位置 0。

·N——负数标志(Negative Flag)。它表示本次运算结果是否为负。它的实质是把本次运算结果的最高位(D7 位)复制到 N 中。(对于带符号的八位二进制数,D7 位为 0 是正数;D7 位为 1 是负数。)

若 V=0(无溢出时),N 代表运算结果的符号:即运算结果为负数时 N 为 1,否则 N 为 0。

若 V=1(有溢出时),则 N 代表的符号与运算结果的符号相反。即运算结果是负数时 N=0,是正数时 N=1。

对标志位的学习要注意掌握以下几点:

(1) 各指令对标志位的影响不尽相同(详见附录)。

(2) 寄存器 P 中各标志位的状态一般是指当前指令执行后的状态,所以标志位常用在执行条件转移时作为判断的依据。

(3) 6527CPU 还可用专门指令对某些标志位进行操作。如“CLC”指令是对 C 位置 0,对其余各位不影响;“CLD”指令是对 D 位置 0,对其余各位不影响。

2·1·3 6527CPU 的复位

6527CPU 的复位地址存在 $FFFC 和 $FFFD 单元中。开机后,系统自动执行由 $FFFC、$FFFD 指定地址的复位程序。F BASIC 工作系统的复位程序的入口一般设为 $C400。

2·1·4 6527CPU 的中断

6527CPU 的中断矢量为 $FFFA～$FFFB 和 $FFFE～$FFFF,其中 $FFFA、$FFFB 为非屏蔽中断 NMI 的地址,中断请求信号由 PPU 的第 19 脚发出,在场回扫的消隐期间触发,中断频率为 50Hz,中断间隔为 0.02 秒。$FFFE～$FFFF 为可屏蔽中断 IRQ 的矢量地址,一般不使用。

2·1·5 6527CPU 管理的内存分布

6527CPU 有 16 位地址线,最大寻址能力为 64K 字节。其内存地址分布情况如表 2—1。

表 2—1　6527CPU 管理的内存地址分布表

地址区间	用　　途
$0000～$00FF	系统 0 页
$0100～$01FF	系统堆栈
$0200～$02FF	卡通定义区
$0300～$07FF	软件中间数据暂存区。其中$0480～$04FF 为功能键定义区
$0800～$1FFF	空区。影射于$0000～$07FF
$2000～$3FFF	与 PPU 相连的 I/O 区
$4000～$5FFF	CPU 的 I/O 区
$6000～$7FFF	8K 空区。根据需要可扩充为 RAM 区。F BASIC 将其作为用户程序存储区
$8000～$FFFF	软件 ROM 区。共 32K 字节。对软件容量大于 32K 的 高 K 软件，在此区间有存储体切换。

2·1·6　6528PPU 管理的内存分布

6528PPU 有 14 位地线，可管理 16K 内存，其管理的内存分布如表 2—2 所示。

表 2—2　6528PPU 所管理的内存分布表

地　址　区	用　　途
$0000～$0FFF	图形库第Ⅰ区，共 4K 空间，存储 256 个 8×8 点阵的卡通图形块。
$1000～$1FFF	图形库第Ⅱ区，共 4K 空间，存储 256 个 8×8 点阵的背景图形、字符图形块。
$2000～$23BF	背景第一页(00 页)屏幕映射区，每一个单元存储一个 8×8 点阵的图形块代码。
$23C0～$23FF	背景第一页(00 页)配色区，每一个单元存储 16 个显示单元的配色代码。
$2400～$27BF	背景第二页(10 页)屏幕映射区。
$27C0～$27FF	背景第二页配色区。
$2800～$2BBF	背景第三页(01 页)屏幕映射区。
$2BC0～$2BFF	背景第三页配色区。
$2C00～$2FBF	背景第四页(11 页)屏幕映射区。
$2FC0～$2FFF	背景第四页配色区。
$3000～$3EFF	空区。映射于背景页。
$3F00～$3F0F	背景页配色代码定义区。其中$3F00 为底背景配色区。
$3F10～$3F1F	卡通页配色代码定义区。
$3F20～$3FFF	空区。映射于$3F00～$3F1F。

2·2　6527CPU 的显示系统

任天堂的游戏画面之所以精彩逼真、内容丰富，远远优于苹果机、中华学习机，关键在于它采用了独特的屏幕显示结构，从而使游戏画面的各种处理快捷而简便。

我们知道，6527CPU 的显示屏幕由四个显示页面叠加而成，自前向后依次为卡通 0 页、背景页、卡通 1 页、底背景页。各页的排列情况如图 2—1 所示。

各个页面的显示可通过系统软开关灵活地控制。在显示图象时，靠前显示页上的内容可以覆盖后面显示页上的内容。因此，通过显示页面间的协调配合可以形成具有一定景深次序的、立体感极强的三维画面。

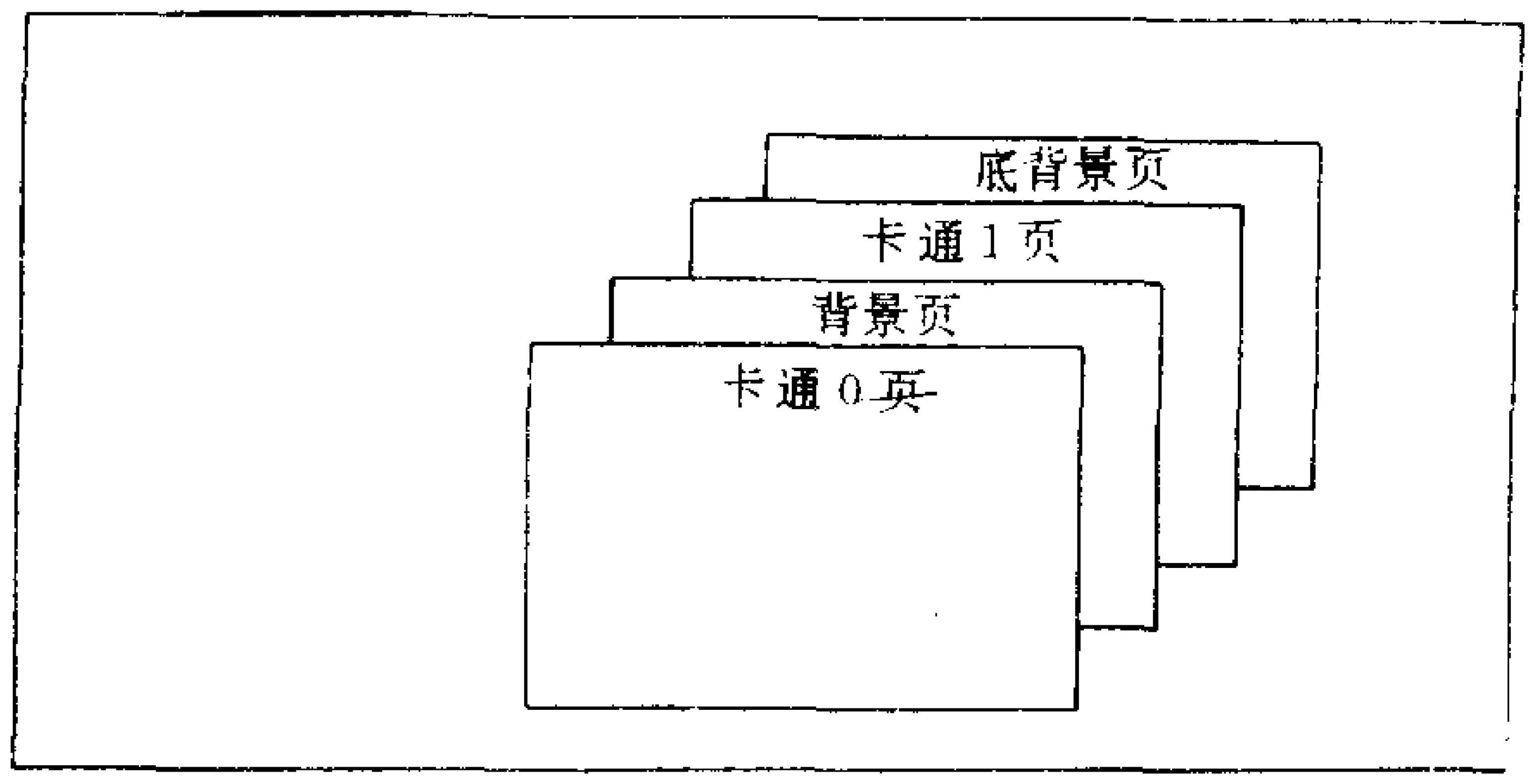


图 2—1　6527CPU 显示页面结构图

2·2·1　背景画面的显示结构

背景页位于卡通 1 页之前、卡通 0 页之后，共由四个显示页面构成。它们的默认排列位置如图 2—2 所示。

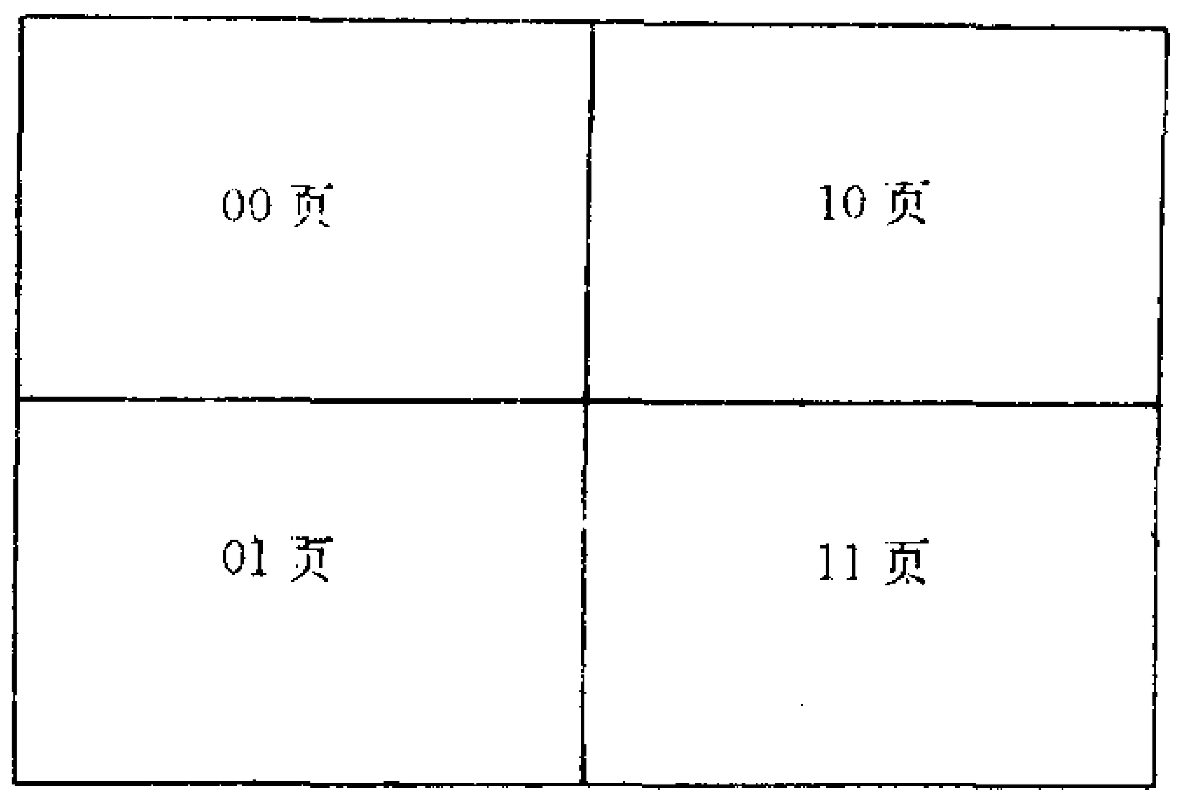


图 2—2　背景页面排列示意图

正常显示时，屏幕显示背景 00 页的内容，工作中可通过系统软开关的设置显示任一页。还可以由系统软开关控制，以点为单位实现两个显示页面的左右卷绕或上下翻滚。

一、显示元素与 PPU 地址的对应关系

在四个背景页面中，每个背景页面都可以显示背景图案或各种字符，每页可显示 32 列×30 行 960 个图形块，由 PPU 直接控制进行定格显示，每个显示元素与 PPU 地址的关系为顺序对应。背景 00 页的显示，使用了 PPU 所管理内存的 \$2000～\$23BF 共 960 个单元，显示行与 PPU 地址的对应关系如表 2—3。背景 01 页的显示使用 \$2400～\$27BF 共 960 个单元。显示行与 PPU 地址的对应关系如表 2—4；每个显示元素与 PPU 地址的对应关系见图 2—3、图 2—4；其它两个页面的对应规律与上面两页相同。但是，由于 PPU 的 VRAM 只有 2K 空间，所以 CPU 自动设定 00、01 页映射到同一 VRAM 单元。即对 00 页显示单元置入数值时，可在 00、01 两页同时显示。同理，当对 01 页显示单元 \$2800～\$2BBF 置入数值时，也可在 00、01 两页同时显示。同样，10、11 两页也映射到同一 VRAM 单元。因此，通常把 00、01 页称为背景 0 页，把 10、11 页称为背景 1 页。特殊需要时，可用软开关把两个页面分开，写入不同的数据以显示不同的图形。

表 2—3　背景 0 页(00 页)各显示行的对应地址一览表

行　次	PPU　地　址	行　次	PPU　地　址
一	2000 ～ 201F	十六	21E0 ～ 21FF
二	2020 ～ 203F	十七	2200～ 221F
三	2040 ～ 205F	十八	2220 ～ 223F
四	2060 ～ 207F	十九	2240 ～ 225F
五	2080 ～ 209F	二十	2260 ～ 227F
六	20A0 ～ 20BF	廿一	2280 ～ 229F
七	20C0 ～ 20DF	廿二	22A0 ～ 22BF
八	20E0 ～ 20FF	廿三	22C0 ～ 22DF
九	2100 ～ 211F	廿四	22E0 ～ 22FF
十	2120 ～ 213F	廿五	2300 ～ 231F
十一	2140 ～ 215F	廿六	2320 ～ 233F
十二	2160 ～ 217F	廿七	2340 ～ 235F
十三	2180 ～ 219F	廿八	2360 ～ 237F
十四	21A0 ～ 21BF	廿九	2380 ～ 239F
十五	21C0 ～ 21DF	三十	23A0 ～ 23BF

表 2—4　背景 1 页(10 页)各显示行的对应地址一览表

行　次	PPU　地　址	行　次	PPU　地　址
一	2400 ～ 241F	十六	25E0 ～ 25FF
二	2420 ～ 243F	十七	2600 ～ 261F
三	2440 ～ 245F	十八	2620 ～ 263F
四	2460 ～ 247F	十九	2640 ～ 265F
五	2480 ～ 249F	二十	2660 ～ 267F
六	24A0 ～ 24BF	廿一	2680 ～ 269F
七	24C0 ～ 24DF	廿二	26A0 ～ 26BF
八	24E0 ～ 24FF	廿三	26C0 ～ 26DF
九	2500 ～ 251F	廿四	26E0 ～ 26FF
十	2520 ～ 253F	廿五	2700 ～ 271F
十一	2540 ～ 255F	廿六	2720 ～ 273F
十二	2560 ～ 257F	廿七	2740 ～ 275F
十三	2580 ～ 259F	廿八	2760 ～ 277F
十四	25A0 ～ 25BF	廿九	2780 ～ 279F
十五	25C0 ～ 25DF	三十	27A0 ～ 27BF

图 2—3、2—4 说明：

(1) 图中的列次表示图形块或字符位于显示屏幕的第几列，行次则表示图形块或字符位于显示屏幕的第几行。如“COMPUTER”的第一个字母“C”则位于第四列第二行。

(2) 每个显示元素所对应的 PPU 行地址加 PPU 列地址即为该显示元素所对应的 PPU 地址。如字母“C”的行地址为 2020H、列地址为 03H，那么“C”的 PPU 地址则为 2023H，记为 $ 2023。其它依次类推。

二、显示单元的配色

为叙述方便，我们把屏幕上的一个图象元素(一个 8×8 点阵的方块)称为一个显示单元，它是背景显示页面的最小显示单位，也可以称为一个图形块。由于每一个显示单元可以显示一个图形块，那么每个显示单元也应该有其确定的配色，也只有对每个显示单元随意置色，才能

行次	PPU行地址 \ 列次	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	2000																																
2	2020				C	O	M	P	U	T	E	R																					
3	2040																																
4	2060																																
5	2080																																
6	20A0																																
7	20C0																																
8	20E0																																
9	2100																																
10	2120																																
11	2140																																
12	2160																																
13	2180																																
14	21A0																																
15	21C0																																
16	21E0																																
17	2200																																
18	2220																																
19	2240																																
20	2260																																
21	2280																																
22	22A0																																
23	22C0																																
24	22E0																																
25	2300																																
26	2320																																
27	2340																																
28	2360																																
29	2380																																
30	23A0																																
	PPU列地址	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F

图 2—3　背景 0 页(00 页)显示单元与 PPU 地址对应图

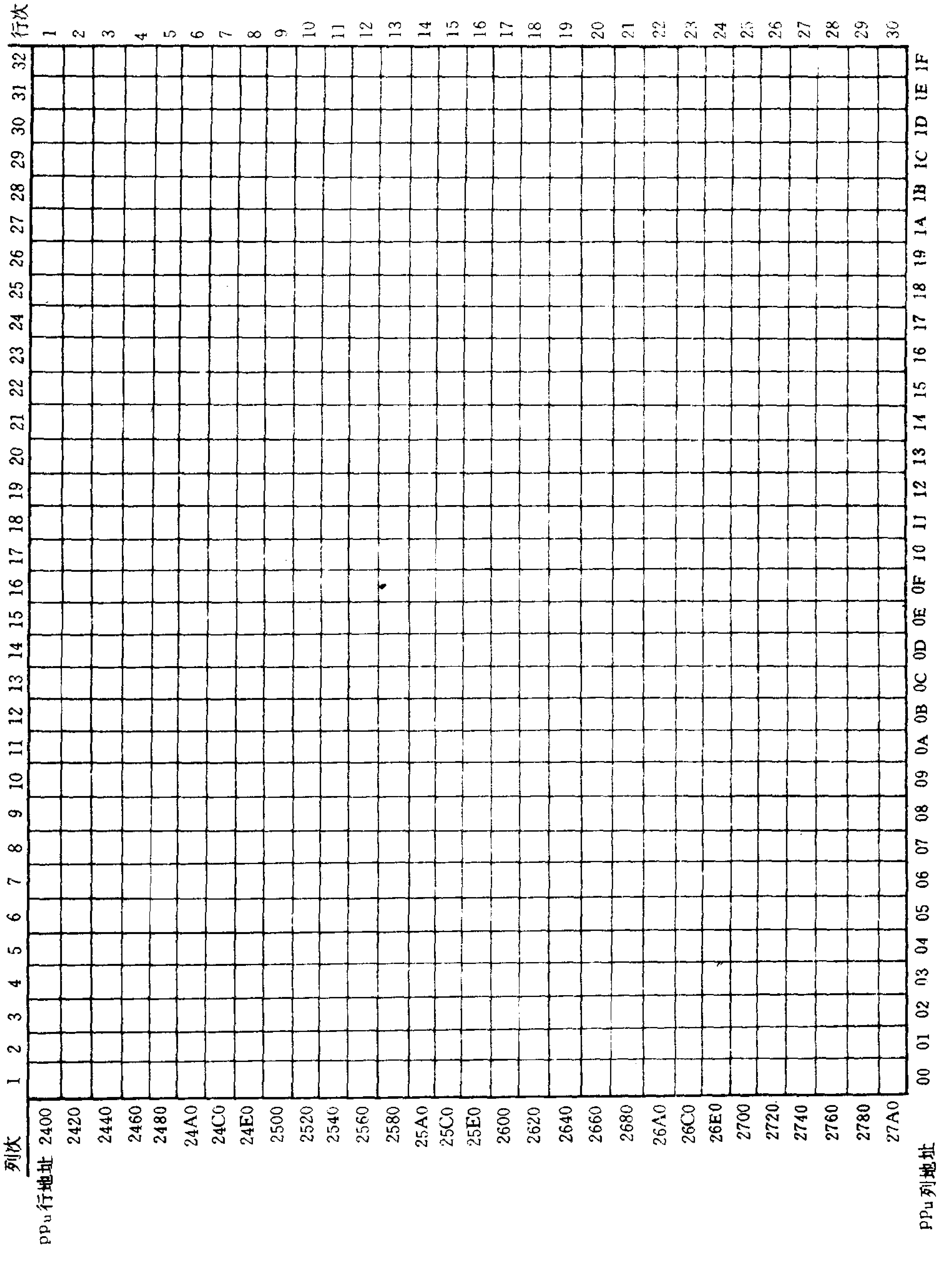


图 2—4　背景 1 页(10 页)显示单元与 PPU 地址对应图

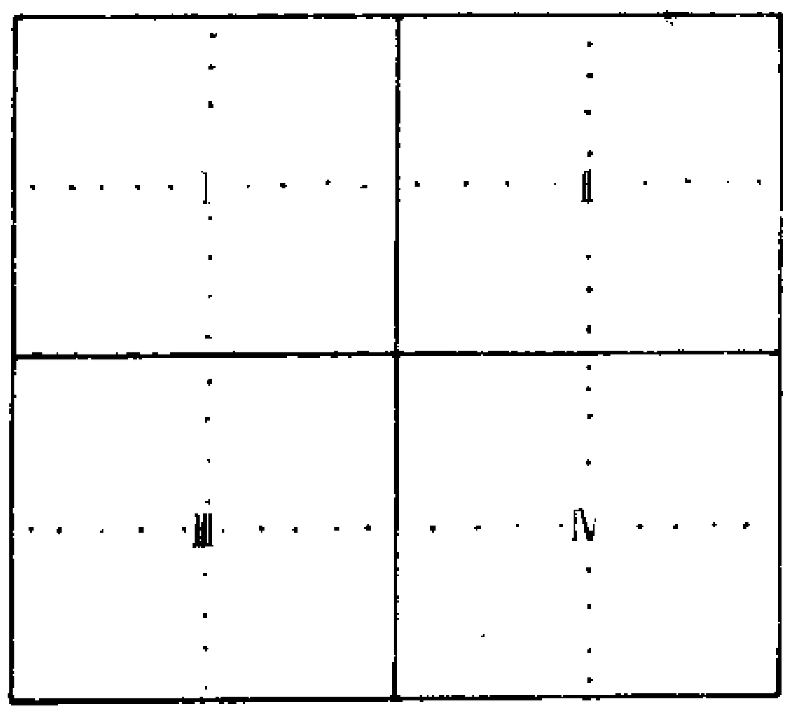
显示出五彩缤纷的绚丽图案。

6527CPU 的配色方法独具特色，它既操作简单又变化千般，是诸多微机系统中较优化的配色方法。它的特点如下：

(1) 它把背景显示页的 32 列×30 行 960 个显示单元按一定格式进行新的排列组合，使每相邻的四个显示单元为一组，把整幅画面分成 16 列×15 行共 240 个小方格(第 15 行每格只有两个显示单元)。每个小方格有四组颜色供选择，这四组颜色对应着 F BASIC 语言中背景配色板中的四个配色代码。由于每个配色代码由三种颜色代码组成，故每个小方格有 12 种颜色供选用(若包括底背景配色，实际有 16 种颜色。)每个小方格的四种配色代码由两位二进制数表示(即显示画面中每相邻的四个单元用一种配色代码)。

(2) 再把 16 列×15 行的 240 个小方格，以上下左右相邻的四个小方格为单位组成一个大方格，即每个大方格由四个小方格、16 个显示单元组成，其配色代码则由 8 位二进制数表示，刚好占用一个配色单元。每个小方格的配色代码在配色单元的排列方式如下图所示。

显示单元分组情况



配色单元各位使用情况	D7	D6	D5	D4	D3	D2	D1	D0
	Ⅳ		Ⅲ		Ⅱ		Ⅰ	

显示单元配色分组及配色代码排列示意图

(3) 通过(1)、(2)两步的处理，把一个显示页面分成 8×8 列 64 个大方格，共占用 64 个配色单元。背景 00 页的配色占用 $23C0～ $23FF；背景 10 页的配色占用 $27C0～ $27FF；01 页的配色占用 $2BC0～ $2BFF；11 页占用 $2FC0～ $2FFF。为便于使用，特把 00、10 页的显示单元与配色地址的对应关系示于图 2－5、图 2－6。

当欲改变某显示单元的配色时，只要把配色数据写入相应的配色单元就可达到目的。因此，6527CPU 显示配色的设置操作是十分简单方便的。

2·2·2 卡通页面的显示结构

卡通页即是游戏中角色所使用的页面，分 0 页和 1 页。显示在 0 页上的角色可以遮住背景物，相当于卡通在背景前面活动；显示在 1 页上的角色则相反，可被背景物所遮住，相当于卡通在背景后面活动。因此，在动画处理中，不须象其它微机那样对背景画面点阵进行复杂的逻辑运算就能达到十分理想的效果。

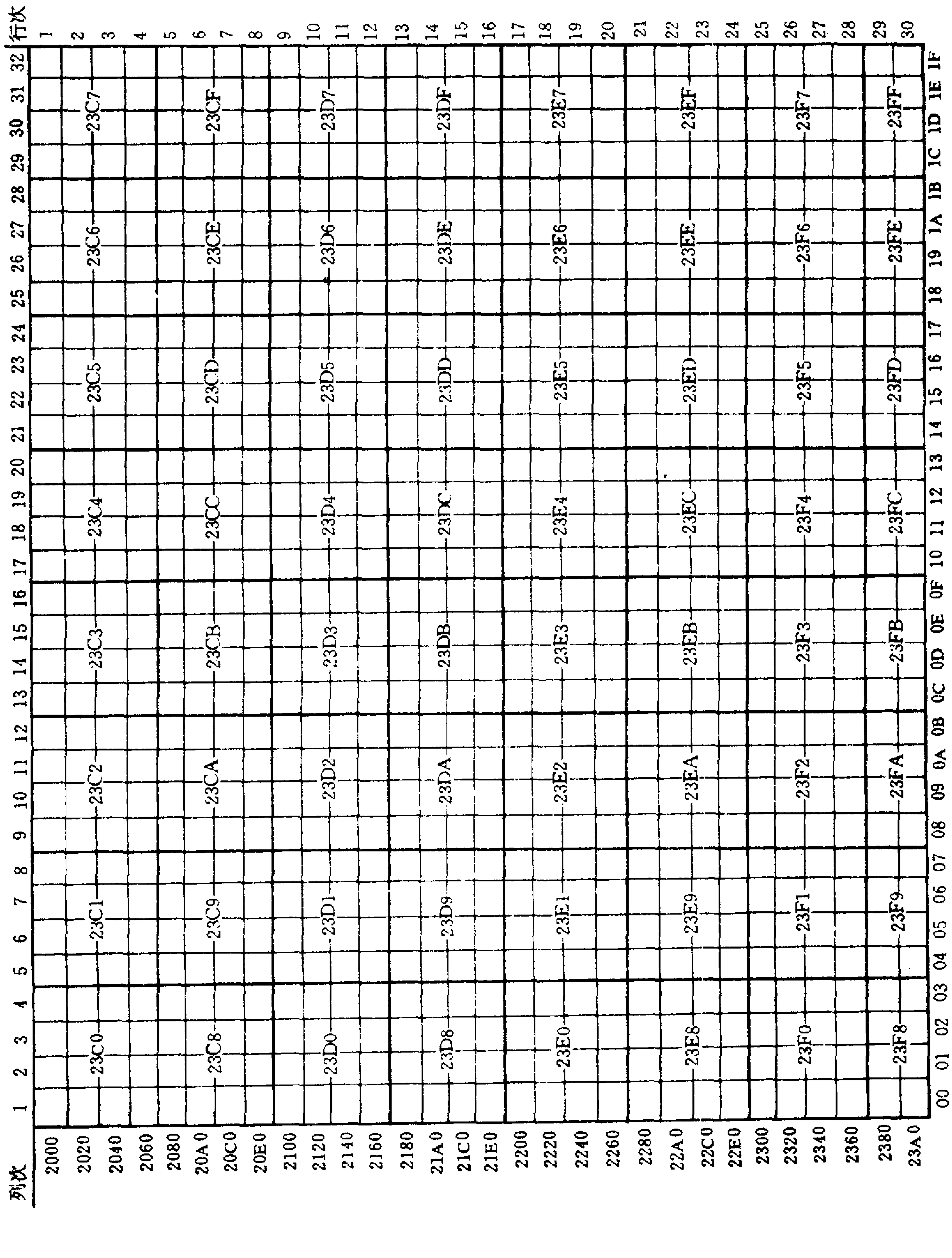


图 2—5 背景 00 页显示单元与配色地址的对应关系示意图

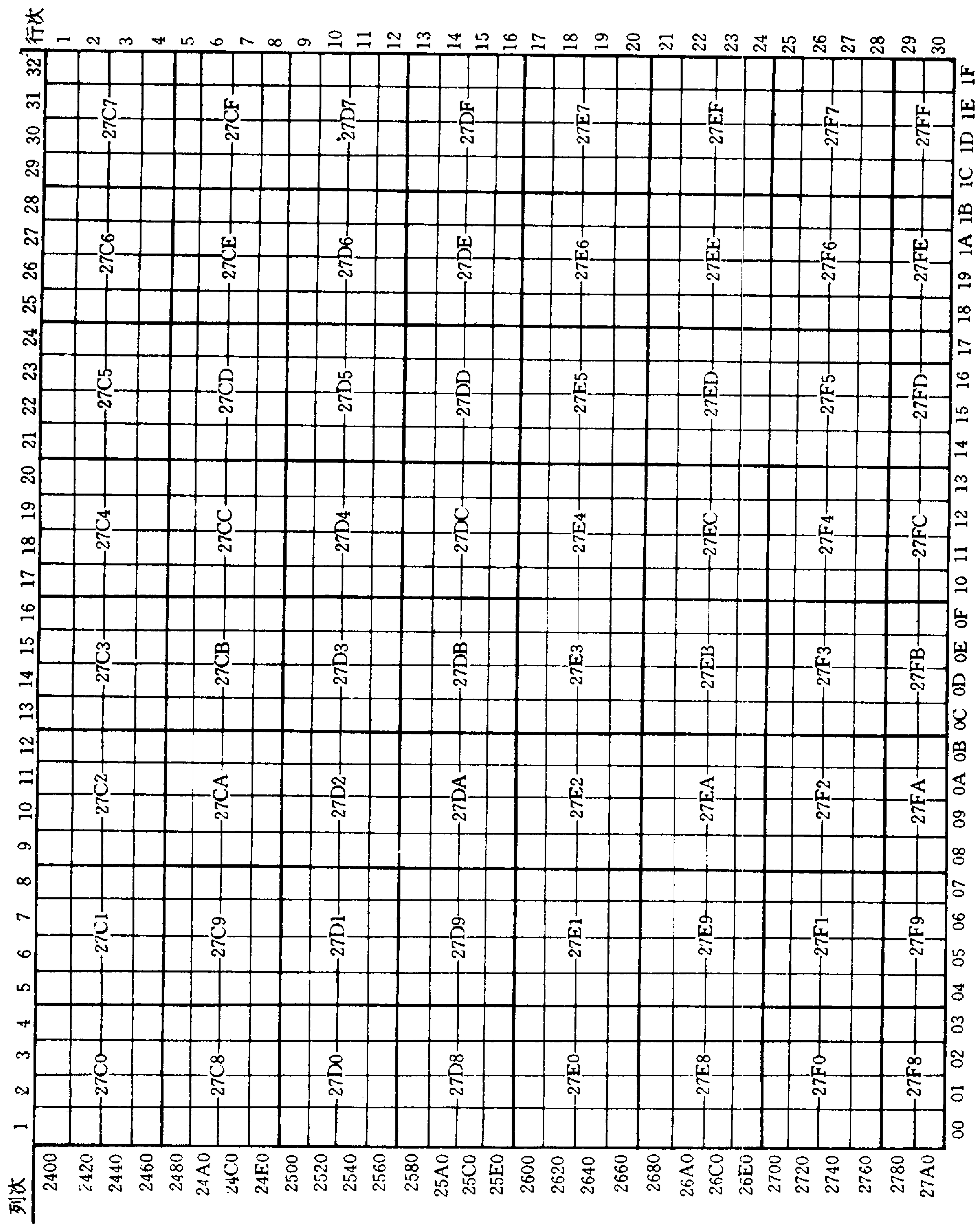


图 2－6　背景 10 页显示单元与配色地址的对应关系示意图

一、卡通页的显示特点

卡通页最多可同时显示 64 个 8×8 点阵角色(卡通块),角色的活动范围为 256×240 点,每次最少可移动一点。可任意指定每个角色左右翻转、上下颠倒等显示状态和四种配色代码的变换。

二、卡通的显示区

卡通页的显示区映射在CPU 所管理内存的第二页中的256 个单元。这样设计有利于对卡通运动控制的快速处理。在＄0200～＄02FF 中,以四个连续单元为一体定义一个 8×8 点阵的卡通块。因此,在卡通定义区最多可定义 64 个卡通块。在 F BASIC 中一般由四个卡通块组成一个 16×16 点阵的角色,所以系统规定最多可显示 16 个角色(SPRITE,简写 SP),即 8 个显示 SP、8 个运动 SP。它们对应的二页地址如表 2—5。

表 2—5 F BASIC 中显示 SP、动作 SP 对应地址表

显示 SP 编号	地址	动作 SP 编号	地址
0	200 ～ 20F	0	280 ～ 28F
1	210 ～ 21F	1	290 ～ 29F
2	220 ～ 22F	2	2A0 ～ 2AF
3	230 ～ 23F	3	2B0 ～ 2BF
4	240 ～ 24F	4	2C0 ～ 2CF
5	250 ～ 25F	5	2D0 ～ 2DF
6	260 ～ 26F	6	2E0 ～ 2EF
7	270 ～ 27F	7	2F0 ～ 2FF

三、显示卡通块的定义

定义一个卡通块要使用二页连续的四个单元,对 64 个卡通块来说,每个对应单元的功能是相同的。下面仅以第一个显示卡通块为例介绍每个单元的含义。(第一个卡通占用＄200～＄203 四个单元)

(1) ＄200:卡通块显示纵坐标设定单元。取值范围为 0～255,有效值为 0～239。

(2) ＄201:卡通块在字库中的序号设定单元。它可以取第一区的序号,也可以取第二区的序号。取值范围为 0～255。

(3) ＄202:卡通块显示状态设定单元。各位的定义如下:

D7	D6	D5	D4	D3	D2	D1	D0

D0、D1:卡通块配色设定位,取值为二进制数 00～11。

D4:卡通类型标志。在 F BASIC 中

D4＝0 表示定义的是显示卡通;

D4＝1 表示定义的是动作卡通。

D5:显示页面标志。(在 F BASIC 中称为显示优先度)

D5＝0,卡通在 0 页显示;

D5＝1,卡通在 1 页显示。

D6:左右翻转标志。D6＝0 为正常显示;

D6＝1 为左右翻转显示。

D7:上下颠倒标志。D7＝0 为正常显示;

D7＝1 为上下颠倒显示。

(4) $ 203:卡通显示横坐标设定单元。取值范围为 0～255。

显然,当欲在屏幕上某一个位置显示某个卡通块时,只要向有关地址写入相应的数据就可任意显示。

例 2－1　在屏幕的(100,150)位置显示黑体数字“4”。

可置入如下数值:

$ 200＝&H96 (150 的 16 进制数)

$ 201＝&HF3 (卡通块“4”的序号为 243,16 进制为 F3H)

$ 202＝0　　(显示状态取正常显示,0＃配色)

$ 203＝&H64 (100 的 16 进制数)

相应的 F BASIC 程序为 No. 2－1

```
5 REM No. 2－1
10 SPRITE ON
20 POKE 512, 150
30 POKE 513, 243
40 POKE 514, 0
50 POKE 515, 100
```

当欲使它左右翻转时,可把 40 句改为

```
40 POKE 514, 64
```

就可实现左右翻转。

同样,若欲使其上下颠倒则改为

```
40 POKE 514, 128
```

若欲使其上下颠倒、左右翻转则改为

```
40 POKE 514, 192
```

若把程序 No. 2－1 改写为 No. 2－2,将会看到数字“4”反复地显示其四种不同的状态,顺序为:正常显示、左右翻转、上下颠倒、上下左右翻转颠倒。

程序 No. 2－2 如下:

```
5 REM No. 2－2
10 SPRITE ON
20 POKE 512, 150
30 POKE 513, 243
40 POKE 514, A
50 POKE 515, 100
60 A = A + 64 : IF A > 192 THEN A = 0
70 PAUSE 50 : GOTO 40
```

若仔细观察将会发现,数字“4”的四种显示状态尽管反复变化,但其右下角的位置恒定不变。这说明 $ 200, $ 203 中指定的坐标点作用在卡通块的右下角。这一点,将为我们在程序设计中准确地确定卡通的显示坐标,提供了遵循的准则。

四、角色的定义方法

BS. 2A 版本图形库中定义的角色一般为 16×16 点阵,即由四个卡通块组成一个 SP。这四

个卡通块的排列顺序如下：

1	2
3	4

显然，当定义一个卡通时，四个卡通块的坐标是不同的。系统规定：16×16 点阵 SP 的基准坐标仍为第一个卡通块的右下角，其余三个卡通块的坐标则以第一个卡通块的坐标为参考点相应调整。因此，当第一个卡通块的坐标为(100,150)时，第二、三、四号卡通块的坐标依次为(108,150)、(100,158)、(108,158)。即以一号卡通块的坐标为参考，二号卡通块的 Y 坐标不变、X 坐标加 8；三号卡通块的 X 坐标不变、Y 坐标加 8；四号卡通块的 X、Y 坐标都加 8。

现以 0# 显示 SP 的定义为例，介绍各单元数据的设置。

0# 显示 SP 占用 $200～$20F 共 16 个单元，当欲在(96,144)位置显示马里欧哥哥时，则应对各单元置入如下数据(假定马里欧哥哥以正常状态显示、0# 配色)：

$200=90　　　$204=90
$201=0　　　$205=1
$202=0　　　$206=0
$203=60　　　$207=68

$208=98　　　$20C=98
$209=2　　　$20D=3
$20A=0　　　$20E=0
$20B=60　　　$20F=68

以上等式中的数据均为 16 进制数。

如果我们对 $200～$20F 单元置入上述数据则与执行 F BASIC 程序

```
××  DEF SPRITE 0，(0，1，0，0，0)≡CHR$(0)+CHR$(1)+ CHR$(
    2)+ CHR$(3)
××  SPRITE 0，96，144
```

取得相同的效果。

通过以上的讨论，我们不难想象：任天堂游戏中的大角色(32×32 点阵或 64×64 点阵)无非是使用较多的卡通块组成。如 32×32 点阵角色的定义，只要用 16 个卡通块按照定义 16×16 点阵 SP 的方法设置坐标就可实现。(在对话工作系统的分析中将会看到，羽毛笔是由 8 个 SP 块组成的定义过程。)

实际上，在任天堂游戏程序的设计中，64 个 8×8 点阵的卡通块根本不够用。解决的方法是：CPU 反复地改写定义卡通块的地址，使几个角色使用同一个映射地址而交替显示。当改写的频率较高时，通过人眼的视觉惰性就能够产生各个角色同时在不同的位置显示的效果。当然，这样必然占用了 CPU 大量的工作时间，这也就是任天堂游戏中大角色移动时产生闪烁的原因。

2·2·3　屏幕显示状态的设置

6527CPU 屏幕显示状态的设置，一般通过两个系统软开关组 $2000、$2001 进行控制。这两个软开关组位于 PPU 内部，可通过编程控制它们的开关状态。现将它们的作用介绍如下：

一、状态方式开关组 $2000

$2000 软开关组主要用于控制 6528PPU 的工作状态：由它的各位决定背景页、卡通页显示所使用的图形库区号；控制向 VRAM 写入数据的方式；选定当前显示的背景页面；控制中断申请等。各位功能见图 2—7。

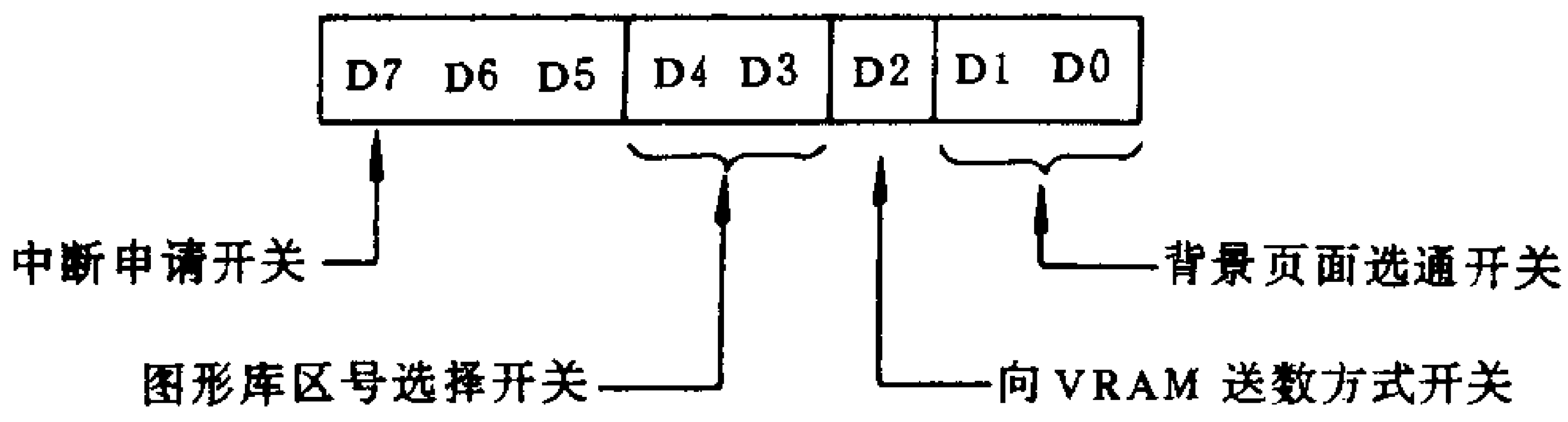


图 2—7 工作状态开关组示意图

图 2—7 说明：

(1) 背景页面选择开关组 D1、D0 位

D1、D0 两位为背景显示页面的选通开关组，两位二进制数可选通四个显示页面。它们的取值与背景页面的对应关系见表 2—6。

表 2—6 $2000 单元 D1、D0 位与背景页面对照表

D1 位	D0 位	选 通 的 背 景 显 示 页 面
0	0	00 页
0	1	10 页
1	0	01 页
1	1	11 页

2、向 VRAM 送数方式选择开关 D2 位

D2 位为向 VRAM 写入数据方式选择开关。当我们向 VRAM 写入数据时，只要向 PPU 置入起始地址，以后可以把欲写入的数据依次送入 VRAM，而不必每写一数据置一次地址，PPU 会控制写入地址自动增加，使地址指针指向下一个置数地址。但是，在写入数据的过程中有一个传数方向问题，表现在屏幕上就是：传送的数据是横向置入在屏幕上显示一行，还是纵向置入在屏幕上显示一列。D2 位就是为实现这一控制而设置的传数方式软开关。

当 D2=0 时，每送入一个数据，PPU 地址自动加一，表现在屏幕上就是横向显示；

当 D2=1 时，每送入一个数据，PPU 地址自动加 20H，表现在屏幕上就是纵向显示。

3、图形库区号选择开关组 D3、D4 位

D3、D4 位为图形库区号选择开关组。其中 D3 位作用于卡通页、D4 位作用于背景页。当它们为 0 时，选中图形库第 I 区的图形，即卡通块的图形，也就是 F BASIC 中符号表 A 的图形；当它们为 1 时，则选中图形库第 Ⅱ 区，即背景字符图形，相当于 F BASIC 中的符号表 B 中的图形。

D3、D4 位共有四种组合，它们刚好对应了 F BASIC 中 CGEN 指令的四种状态，其对应关系见表 2—7。

通过表 2—7 的对照，我们可以深刻地体会 CGEN 指令的执行实质。

4、中断申请开关 D7 位

D7 位为中断申请开关。当置 1 时发出中断申请；置 0 时则关闭中断申请。

表 2－7　$ 2000 单元 D4、D3 位与 CGEN 指令状态对照表

D4 位	D3 位	选中的图形库区号		相对应的 CGEN 指令
		背 景 页	卡 通 页	
0	0	Ⅰ	Ⅰ	CGEN0
0	1	Ⅰ	Ⅱ	CGEN1
1	0	Ⅱ	Ⅰ	CGEN2
1	1	Ⅱ	Ⅱ	CGEN3

二、显示开关组 $ 2001

$ 2001 是屏幕显示控制开关组，各位开关的作用见图 2－8。

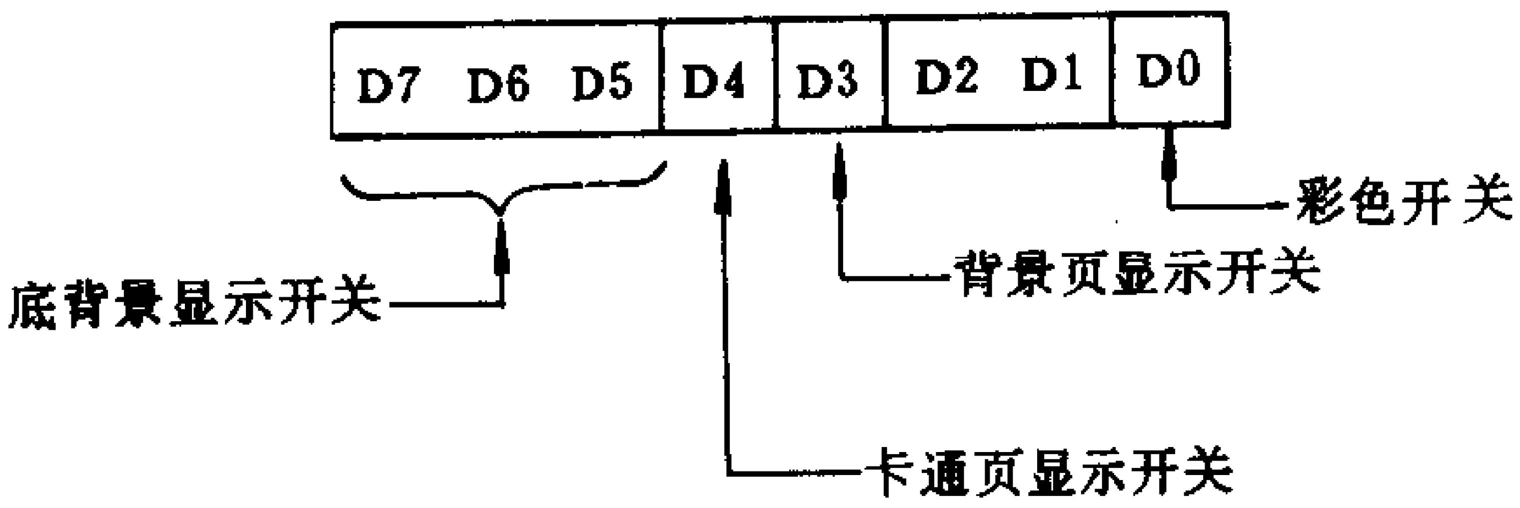


图 2－8　显示开关组功能示意图

说明：

(1) 彩色控制开关 D0 位

DO 位为彩色控制开关。当 D0＝1 时关闭彩色，屏幕显示呈黑白图象。实质上，D0＝1 时关闭了 PPU 内部的彩色发生器；当 D0＝0 时则打开了彩色发生器。

(2) 背景页显示开关 D3 位

D3 位是背景页显示开关。通常 D3＝1，仅当与 PPU 通信时，为避免屏幕闪烁，才令 D3＝0，关闭背景页显示。通信结束后再令 D3＝1，打开显示。当 D3＝0 时光标也将不再显示。

(3) 卡通页显示开关 D4 位

D4 位是卡通页显示开关。D4＝1 时，相当于执行 F BASIC 语句中的 SPRITE ON 指令；D4＝0 则相当于执行 SPRITE OFF。

(4) 底背景显示开关组 D5～D7 位

D5～D7 位为底背景显示开关组。其中 D5 位决定底背景的亮度，D5＝0 时亮、D5＝1 时暗；D6、D7 位可调整底背景的色调，当 D6＝1 时，色调偏暖，D7＝1 时色调偏冷。有了这三个软开关，我们可在不改变底背景配色的情况下，微调底背景的八种显示色彩。

2・2・4　CPU 与 PPU 之间的通信方法

我们知道 PPU 是图象处理器，它与 CPU 分别管理一定的内存单元。因此，它们之间的通信则是 6527 工作系统中至关重要的一环。由电路图可以看出，CPU 有三根地址线 A0～A2 与 PPU 直接相连，这就决定了 CPU 只能通过八个软开关与 PPU 打交道。这八个软开关的对应地址为 $ 2000～ $ 2007，CPU 正是通过这八个软开关的状态设置来控制 PPU 的各种操作的。

以上我们曾讨论了软开关组 $ 2000、$ 2001，常用的软开关还有屏幕位移开关 $ 2005、设置 PPU 通信地址开关 $ 2006 和 PPU 读、写控制开关 $ 2007。下面分别介绍它们的一般使用方法。

一、读取 PPU 内存的方法

读 PPU 内存的汇编语言程序见 No. 2－3。

No. 2－3 读 PPU 内存

```
    LDA  # $00
    STA  $2001     ;关显示以避免屏幕闪烁
    LDA  # $YY     ;YY 为欲读取的 PPU 地址高位
    STA  $2006
    LDA  # $XX     ;XX 为欲读取的 PPU 地址低位
    STA  $2006     ;以上设置了 PPU 的读数首地址 $YYXX
    LDA  $2007     ;触发 PPU 读写开关
    LDA  $2007     ;读取 PPU 数据
    PHA            ;读出的数据入栈保存
    LDA  # $0E
    STA  $2001     ;开显示，0EH 值可根据需要改变
    PLA            ;从堆栈中弹出所读的 PPU 数据，数据存在 A 寄存器中。
    RTS
```

说明：

程序 No. 2－3 只是读取 PPU 内存的原理程序，实际应用时，首先要保存有关寄存器中的原值(即保护现场)；然后设置计数器，累计读数的个数；读数完毕还要恢复现场等。下面给出读取 PPU 的实用程序。

例 2－2　读出 PPU 内存中 $00～$10 单元的 17 个数据存入 $7800～$7810。

实现程序见 No. 2－4。

No. 2－4 读 PPU 内存实用程序

```
     PHA            ;保存 A 寄存器原值
     TXA
     PHA            ;保存 X 寄存器原值
     LDX  # $00     ;计数器清零
     LDA  # $00
     STA  $2001     ;关显示
     LDA  # $00
     STA  $2006
     LDA  # $00
     STA  $2006     ;设置 PPU 首址 $0000
     LDA  $2007     ;触发 PPU 读、写开关
READ LDA  $2007
     STA  $7800,X   ;读数送 $7800+X 单元
     INX
     CPX7 # $11     ;检查取数个数是否达到要求
     BNE READ       ;没达到再读
     LDA  # $0E
     STA  $2001     ;开显示
```

```
PLA
TAX          ;恢复 X 寄存器原值
PLA          ;恢复 A 寄存器原值
RTS
```

执行 No. 2—4,即可把 PPU 的 $00～$10 的数据读出存入 $7800～$7810。

程序 No. 2—4 虽然已经很完美,但是在我们的学习机中还不能直接运行。下面介绍在 F BASIC 状态下使用程序 No. 2—4 的方法。

1、首先把 No. 2—4 汇编成机器语言,假定我们把这段机器语言程序放在用户程序区的最后一页,从 $7F00 开始存放,则 No. 2—4 的机器语言程序如 No. 2—5。

No. 2—5

```
7F00  48        PHA
7F01  8A        TXA
7F02  48        PHA
7F03  A2 00     LDX #$00
7F05  A9 00     LDA #$00
7F07  8D 01 20  STA $2001
7F0A  A9 00     LDA #$00
7F0C  8D 06 20  STA $2006
7F0F  A9 00     LDA #$00
7F11  8D 06 20  STA $2006
7F14  AD 07 20  LDA $2007
7F17  AD 07 20  LDA $2007
7F1A  9D 00 78  STA $7800,X
7F1D  E8        INX
7F1E  E0 11     CPX #$11
7F20  D0 F5     BNE $7F17
7F22  A9 0E     LDA #$0E
7F24  8D 01 20  STA $2001
7F27  68        PLA
7F28  AA        TAX
7F29  68        PLA
7F2A  60        RTS
```

2、把程序 No. 2—5 写入对应的内存单元

把数据写入内存单元有两种方法:

• 一种使用 POKE 指令,把数据逐个写入内存。这种方法简单易行,适用于程序较短的情况,如程序 No. 2—1。

• 第二种方法是编写 F BASIC 循环程序,把数据写入相应的内存。

现在介绍第二种写入方法。写入程序见 No. 2—6。

```
5 REM No. 2—6
10 DATA 72, 138, 72, 162, 0, 169, 0, 141, 1, 32, 169, 0, 141, 6, 32, 169, 0, 141,
```

```
6, 32, 173, 7, 32, 173, 7, 32. 157, 0, 120, 232, 224, 17, 208, 245, 169, 14, 141,
1, 32, 104, 170, 104, 96
20 FOR I = 0 TO 42 : READ A : POKE &H7F00 + I , A : NEXT
30 CALL &H7F00
40 FOR I = &H7800 TO &H7810 : PRINT PEEK ( I ) " " ; : NEXT
```

运行程序 NO. 2－6,打印结果如下:

```
00  00  07  31  63  31  15
48  03  15  00  02  04  30
00  03  00
```

改变 No. 2－6 中的 PPU 地址,可打印出其它地址的数据。

3、对于不懂机器语言的朋友,也可以编写 F BASIC 程序读出 PPU 内存。程序见 No. 2－8。

```
5 REM No. 2－8
10 POKE &H2001 , 0
20 POKE &H2006 , 0
30 POKE &H2006 , A
40 B = PEEK ( &H2007 )
50 B = PEEK ( &H2007 ) : PRINT HEX$ ( B ),
60 A = A + 1 : IF A = 18 THEN 80
70 GOTO 20
80 POKE &H2001 , 14
90 END
```

运行 No. 2－8 将打印出与 No. 2－7 相同的结果。但是,若比较两个程序的运行时间,将会发现程序 No. 2－7 几乎是立即打印出结果,而运行 No. 2－8 则要关闭显示一段时间后,才显示出运行结果。由此可见,机器语言程序的运行速度要比 BASIC 语言快许多倍。

二、写入 PPU 内存的方法

写入 PPU 内存的编程思路和方法与读 PPU 基本相同。汇编语言程序见 No. 2－9。

No. 2－9 写入 PPU 内存

```
LDA  # $00
STA  $2001          ;关显示
LDA  # $YY
STA  $2006
LDA  # $XX
STA  $2006          ;置 PPU 首址
LDA  # $MM
STA  $2007          ;把数据写入 PPU
LDA  # $0E
STA  $2001          ;开显示
RTS
```

一旦设置好 PPU 的起始地址后,可连续写入多个数据。关于机器语言的写入程序,读者可

参照程序 No. 2－6、No. 2－7 自己编写。下面仅给出使用 F BASIC 语言编程写入 PPU 内存的方法。

例 2－3　从背景 0 页的第一行起，显示背景图形库(第二区图形)的 256 个字符、图形。

实现程序见 No. 2－10

```
5 REM No. 2－10
10 POKE 8193 , 0
20 POKE 8192 , 148
30 FOR I = 0 TO 255
40 POKE 8198 , 32
50 POKE 8198 , I
60 POKE 8199 , I
70 NEXT
80 POKE 8193 , 14
90 END
```

运行程序，将打印出 32 列×8 行的 256 个字符和图形。

程序 No. 2－10 提供了解决引言中第一个问题的有效手段。运用 No. 2－10 的编程思路，我们完全可以在 F BASIC 状态下设计出 32 列×30 行的游戏画面。

2·2·5　背景画面的位移

背景画面位移一般分为横向位移、纵向位移和斜向位移三大类。通常称横向位移为画面的左右卷动、纵向位移为画面的上下滚动。多数 CPU 的位移操作是通过对屏幕映射单元不停地改变数据实现的。而 6527CPU 的位移方法则另辟蹊径，采用了屏幕显示映射单元的内容不变，通过移动屏幕显示窗口实现画面位移的方法。这犹如照相机或摄像机中的取景框，显示窗口对准了哪一块图形，就把哪一块图形显示在屏幕上。因此，背景画面的位移实质就是显示窗口向相反的方向移动。由于显示窗口可任意移动，因此 6527CPU 的背景画面可方便地进行任意方向的移动(一般取八个方向)。由于每一个方向的位移可分为往复、循环两种过程，所以 6527CPU 一般有 16 种位移方式。

繁多的位移方式和简单的位移操作是 6527CPU 的突出特点。

下面分别介绍各类位移操作的原理和实现方法。

一、屏幕显示窗口

1、显示窗口的概念

屏幕显示窗口是一个形象的称谓。实际上，它是由卡通 0 页、卡通 1 页、底背景页三层画面叠加而成的一个具有立体结构的画面。因此，显示窗口的平面尺寸为 32 列×30 行。为方便讨论，我们把它抽象为一个具有 32 列×30 行的透明矩形，在示意图中以双线框表示，讨论中简称为显示窗口或直接称窗口。

虽然画面位移的实质是显示窗口移动，但为了符合画面位移的通常概念和称谓习惯，我们在讨论中仍称背景画面位移，但在理论上要清楚实际是显示窗口在移动。

2、显示窗口的坐标系

为使显示窗口的坐标与卡通显示坐标相对应，我们把窗口的坐标原点选在左上角，如图 2－9 所示，坐标值按点阵计算，则 XMAX＝256、YMAX＝240。实用中 X 取值 0～255、Y 取值 0

～239。

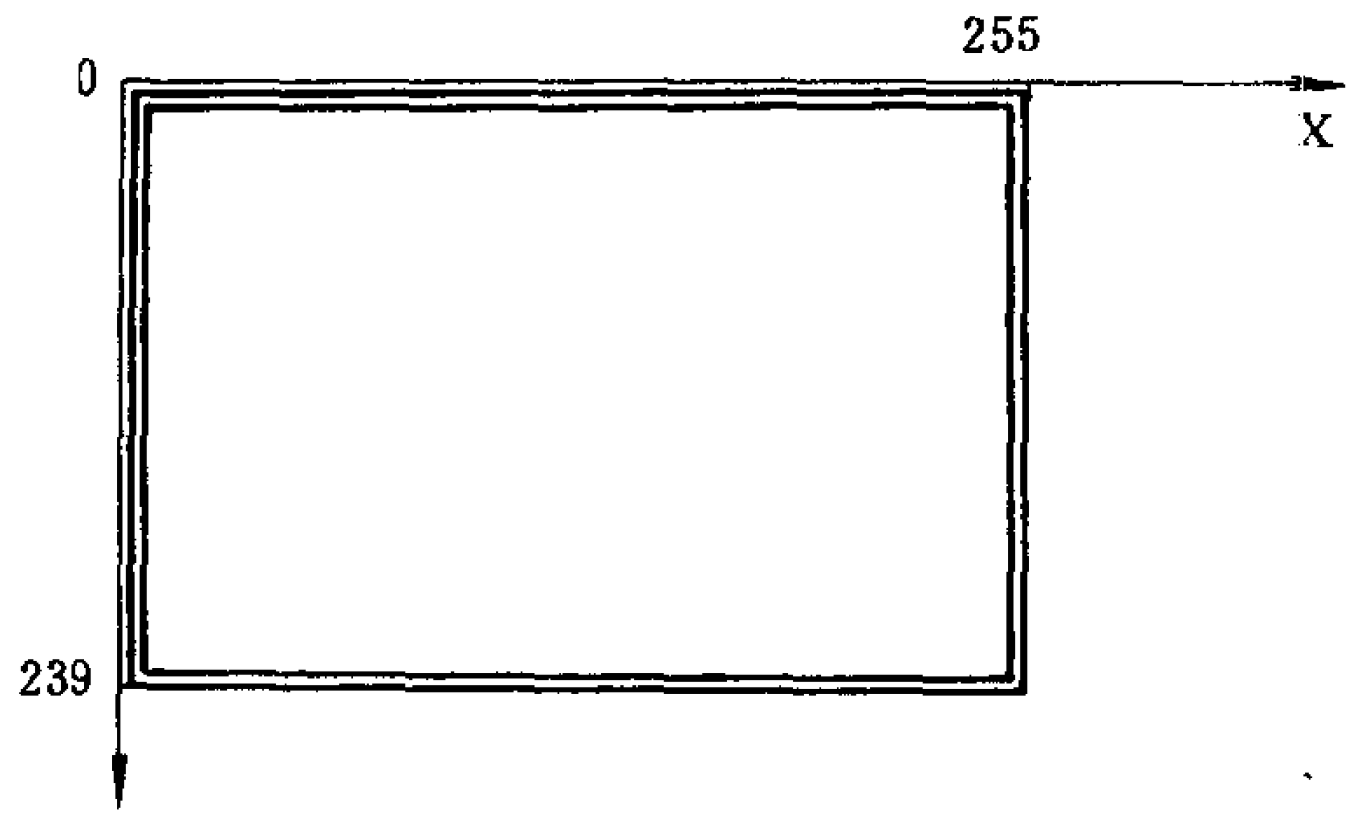


图 2－9　显示窗口坐标系

3、显示窗口与背景画面的对应关系

由于画面的移动，实质上是窗口向相反方向的位移，那么对于图 2－9 选定的窗口坐标系，背景画面与显示窗口将有图 2－10(a)所示的对应关系。

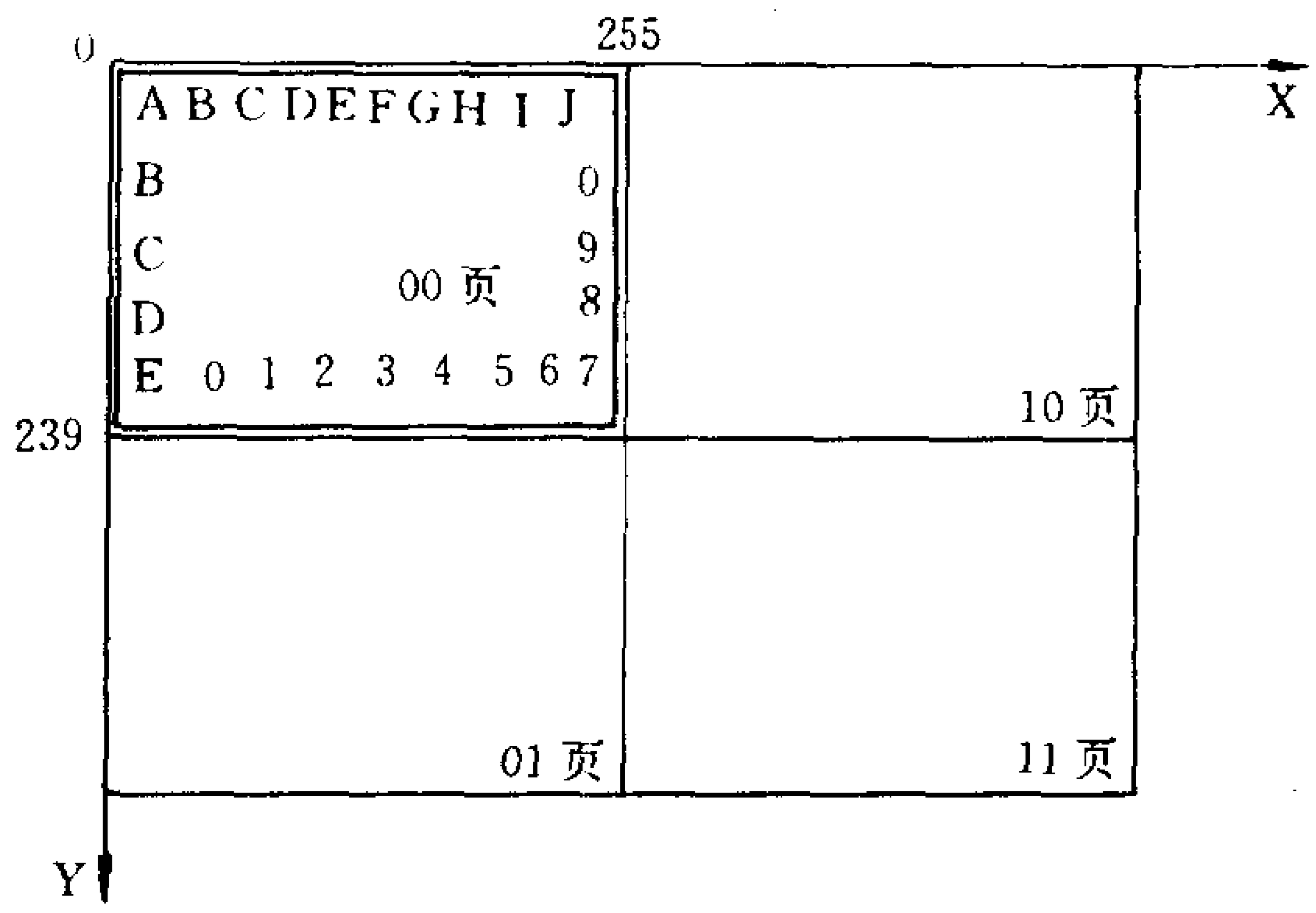


图 2－10(a)　显示窗口与背景画面的关系图

二、画面的简单位移

1、画面的横向位移

画面横向位移的示范程序如 No. 2－11a。

(以移动 a 点为例)

No. 2－11a

```
LDA  # $a
STA  $2005        ;窗口横向位移
LDA  # $00
STA  $2005        ;窗口纵向位移
JSR YANSHI        ;延时
RTS
```

执行 No. 2－11 将有图 2－10(b)的画面。

程序 No. 2－11a 的 F BASIC 程序如 No. 2－11b 所示。

```
5 REM No. 2－11b
10 POKE 8197 , A
20 POKE 8197 , 0
```

```
30 PAUSE 1000 : END
```

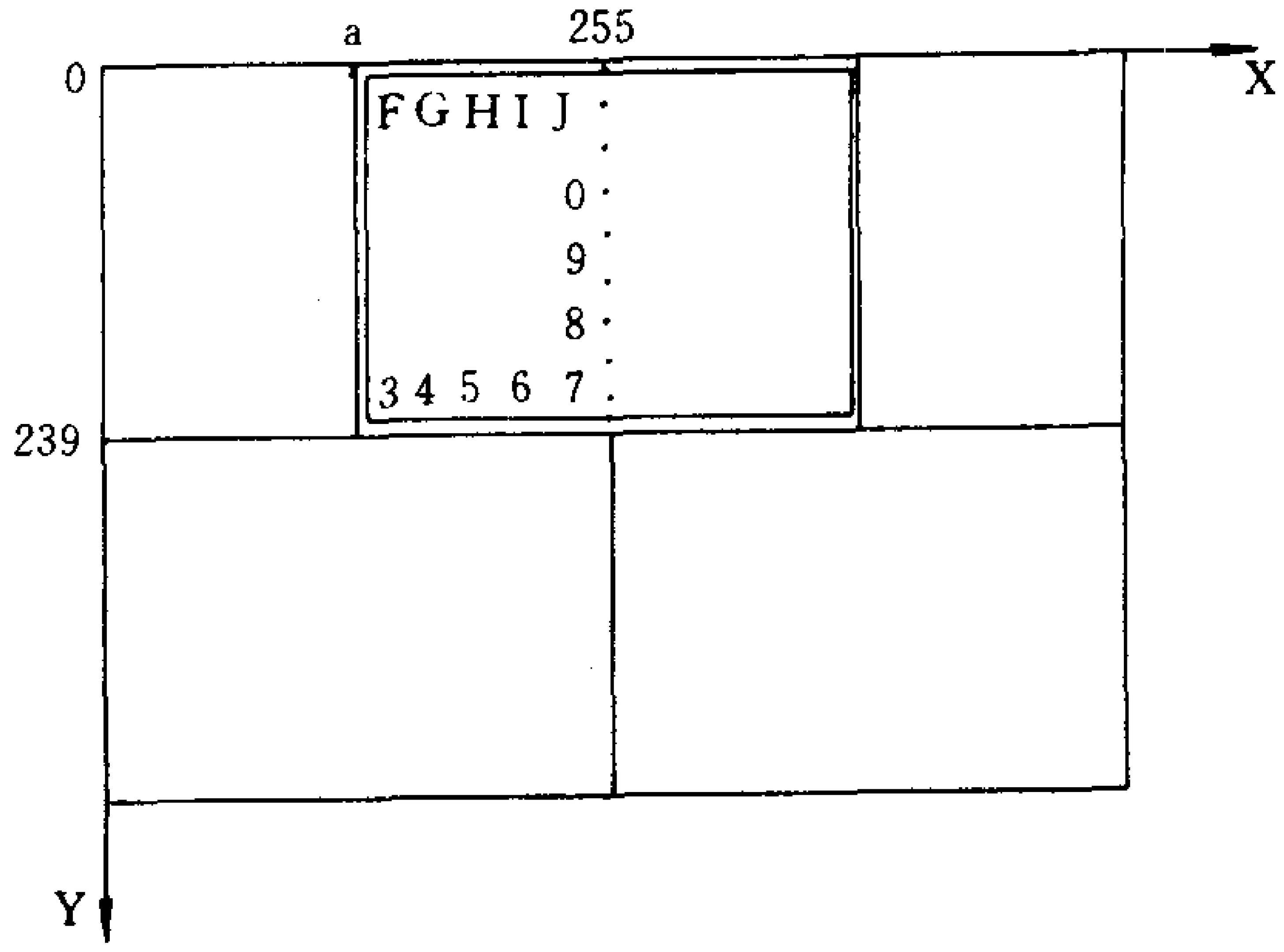


图 2—10(b)　画面横向移动 a 点示意图

2、画面的纵向位移

若令画面纵向位移 b 点，其实现程序如 No. 2—12a。

No. 2—12a

```
LDA #$00
STA $2005
LDA #$b
STA $2005
JSR YANSHI
RTS
```

执行 No. 2—12a，将有图 2—10(c)的画面。

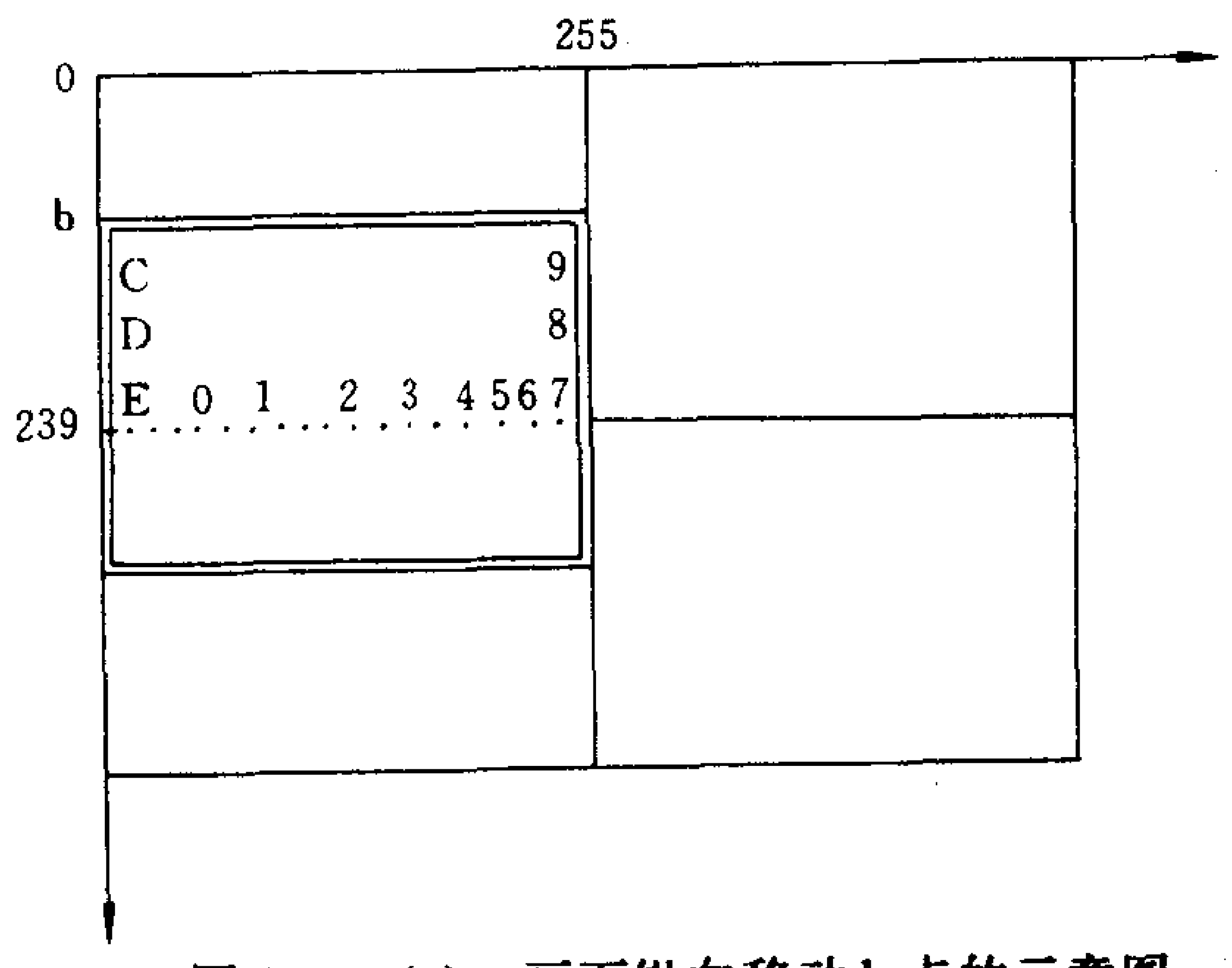


图 2—10(c)　画面纵向移动 b 点的示意图

对应的 F BASIC 程序可比照 No. 2—11b 写出(从略)。

3、画面的斜向位移

若令画面横向位移 c 点、纵向位移 d 点，其实现程序如 No. 2—13。

No. 2—13

```
LDA #$c
STA $2005
LDA #$d
```

```
STA  $2005
JSR YANSHI
RTS
```

执行 No. 2－13，将有图 2－10(d)的画面。

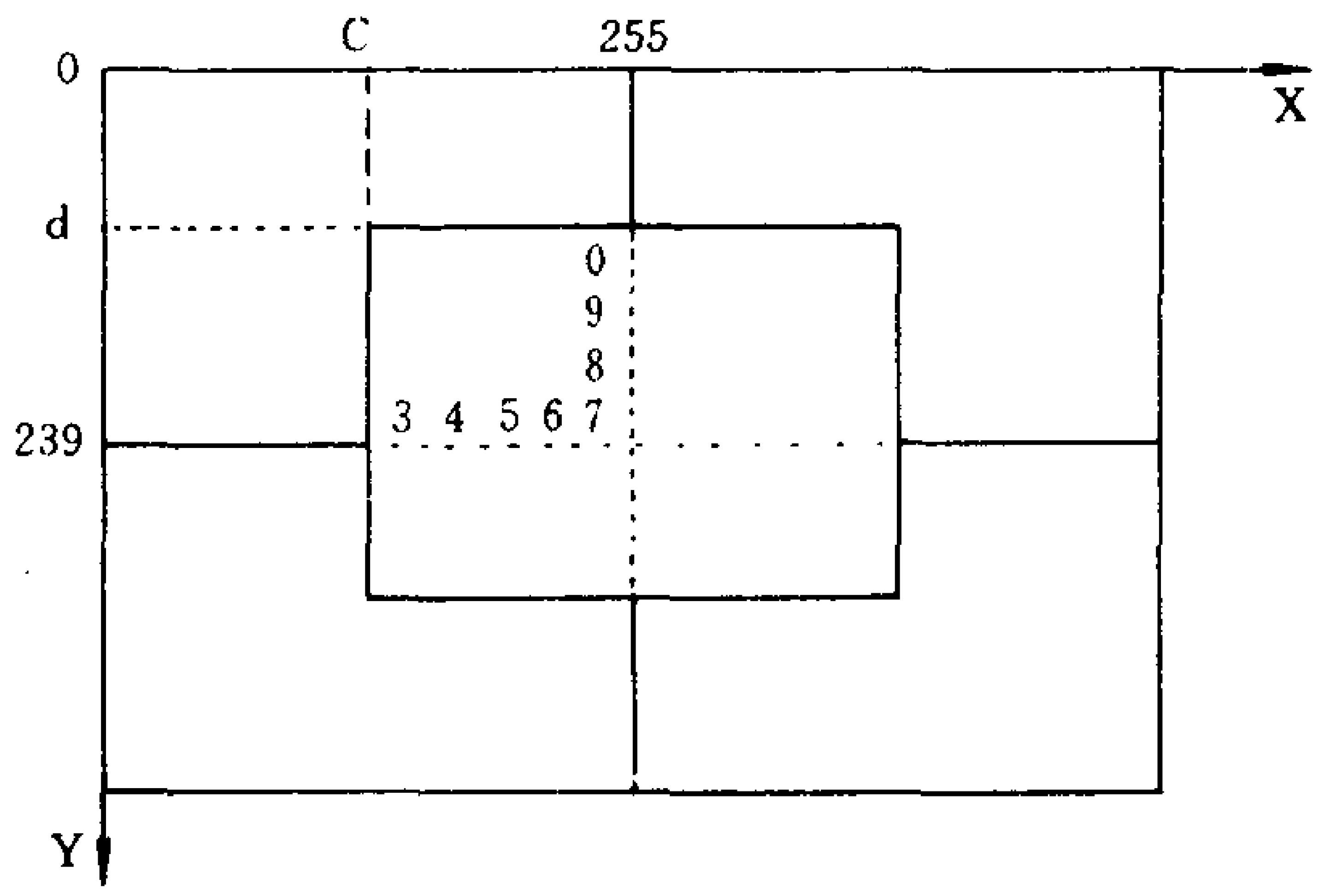


图 2－10(d)　画面斜向位移示意图

通过以上的位移过程可见，6527CPU 的画面移动操作是极为简便的。

三、画面的连续位移

实际游戏中，画面的位移操作多数是以点为单位的连续位移。如《碰碰车》游戏是连续的横向位移；《拆屋工》是连续的纵向位移；《淘金者》、《沙罗曼蛇》等游戏则是横向、纵向连续分段位移；而大多数游戏如《魂斗罗》、《松鼠大作战》、《上尉密令》、《人间兵器》等游戏则是横向、纵向混合位移。从而使游戏画面广阔、博大、丰富迷人。

一般把画面的横向连续位移称为横向卷动，把画面的纵向连续位移称为纵向滚动。下面分别讨论各类连续位移的实现方法。

1、画面连续左移

从图 2－10(b)可以看出，画面的横向位移是在 00、10 页或 01、11 显示页面间的移动。根据 No. 2－11a，我们很容易写出画面连续左移的程序 No. 2－14a。

```
No. 2－14A
     LDX  #$00
     LDY  #$00
LEFT STX  $2005
     STY  $2005
     JSR YANSHI
     INX
     BNE LEFT
     RTS
```

对应的 F BASIC 程序见 No. 2－14b。

```
5 REM No. 2－14b LEFT MOVE
10 A = 0
20 POKE 8197 , A
30 POKE 8197 , 0
```

```
40 A = A + 1 : IF A = 256 THEN PAUSE 1000 : END
50 PAUSE 5
60 GOTO 20
```

执行程序 No. 2—14 可把画面的 00 页连续左移出窗口，使窗口对正 10 页。

调整 50 行 PAUSE 语句的参数，可控制画面左移的速度。

2、画面的连续右移

画面的连续右移程序见 No. 2—15a。

```
No. 2—15a RIGHT MOVE
      LDX # $FF
      LDY # $00
RIGHT STX $2005
      STY $2005
      JSR YANSHI
      DEX
      BNE RIGHT
      RTS
```

对应的 F BASIC 程序见 No. 2—15b。

```
5 REM No. 2—15b RIGHT MOVE
10 A = 255
20 POKE 8197 , A
30 POKE 8197 , 0
40 A = A - 1 : IF A = -1 THEN END
50 PAUSE 5
60 GOTO 20
```

画面连续右移时，首先把窗口迅速移到 10 页面，然后连续右移，直到把 00 页全部移入显示窗口而停止。

3、画面的连续上移

由图 2—10(c)可见，画面的纵向位移是在 00、01 或 10、11 页面间移动。画面连续上移的程序见 No. 2—16a。

```
No. 2—16a UP MOVE
    LDX # $00
    LDY # $00
UP  STX $2005
    STY $2005
    JSR YANSHI
    INY
    CPY # $F0
    BNE UP
    RTS
```

对应的 F BASIC 程序见 No. 2—16b

```
5 REM No. 2－16b UP MOVE
10 A = 0
20 POKE 8197 , 0
30 POKE 8197 , A
40 A = A + 1 : IF A = 240 THEN PAUSE 1000 : END
50 PAUSE 2
60 GOTO 20
```

4、画面的连续下移

画面的连续下移程序见 No. 2－17。

No. 2－17a DOWN

```
     LDX # $00
     LDY # $F0
DOWN STX $2005
     STY $2005
     JSR YANSHI
     DEY
     BNE DOWN
     RTS
```

对应的 F BASIC 程序见 No. 2－17b。

```
5 REM No. 2－17b DOWN
10 A = 240
20 POKE 8197 , 0
30 POKE 8197 , A
40 A = A － 1 : IF A = －1 THEN END
50 PAUSE 2
60 GOTO 20
```

同画面的连续右移的情况相同，下移前显示窗口要迅速跳到01页，然后逐点上移到00页，直到窗口与00页对正后，位移过程结束。

5、画面的横向往复位移

《淘金者》二代游戏中，按暂停键后，可用左右方向键控制画面左右往复运动。这种位移过程的控制程序实际上是把程序 No. 2－14、No. 2－15合并组成的。合并后的程序见 No. 2－18a。

No. 2－28a

```
     LDX # $00
     LDY # $00
LEFT STX $2005
     STY $2005
     JSR YANSHI
     INX
     BNE LEFT
     DEX
```

```
RIGHT STX $2005
      STY $2005
      JSR YAN SHI
      DEX
      BNE RIGHT
      BEQ LEFT
```

对应的 F BASIC 程序为 No. 2－18b。

```
5 REM No. 2－18b
10 A = 0
20 POKE 8197 , A
30 POKE 8197 , 0
40 IF B = 0 THEN A = A + 1 : IF A = 256 THEN B = 1
50 IF B = 1 THEN A = A － 1 : IF A = 0 THEN B = 0
60 PAUSE 5
70 GOTO 20
```

运行程序 No. 2－18就可看到画面先连续左移到背景10页；然后又连续右移到背景00页，不停地往复移动，直到按 STOP 键才停止位移。

为使画面的位移过程更易分辨直观，我们可以在执行 No. 2－18之前，把背景00、10两页都写入内容。背景0页常用的写入方法是在程序中加一条输出语句，向00页写入字符或图形。背景1页的写入方法有两种：一种是直接把字符数据写入 PPU；第二种方法是在 BG GRAPHIC 画面绘制图案或字符(因为 BG GRAPHIC 画面使用背景10页面)。

10页面最简单的绘图方法就是使用 BG GRAPHIC 画面键控绘图。操作过程是：在 BS. 2A F BASIC 状态下，键入 SYSTEM 命令，即返回选项画面；选择2，进入 BG GRAPHIC 绘图状态，进行键控绘图；绘图结束后，返回 F BASIC 状态，在程序 No. 2－18b 中加入

```
15 CLS : FOR I = 0 TO 600 : PRINT " A " ; : NEXT
```

再运行 No. 2－18b 就可以直观、清晰地观察画面的往复位移过程。

另外，在汇编语言位移程序中，我们调用了一个延时子程序"YAN SHI"，可能朋友们编写的延时子程序延时量太小或延时程序较大，使用起来不太方便。现给出一个延时量任意可调的通用延时子程序。

```
        SUB
YANSHI LDA #$a
        STA $401
        LDA #$b
        STA $400
        JSR nn
        RTS
```

程序中 a、b 为延时参数。其中 b 为延时数据低位，b 的取值延时量与 PAUSE 语句中的参数相同；a 为延时数据高位，a 值的一个单位相当于 PAUSE 256。故一般取 a＝0、b＝2～5。"nn"为系统延时子程序的入口地址，对于 BS. 2A 版本的 F BASIC nn＝$91B0；对于 V3·10版本的 F BASIC nn＝$9731。使用"YANSHI"子程序时，只要根据程序延时的要求，用相应的数值代

入 a、b 中就可任意调整延时时间。

6、画面的纵向往复位移

由于画面的纵向往复位移仅在背景0页或1页的本页之间进行，所以其实现程序更为简单。借鉴横向往复位移的编程方法，将很容易写出画面往复位移的实现程序。

No. 2－19a

```
      LDX # $00
      LDY # $00
UP    STX  $2005
      STY  $2005
      JSR YAN SHI
      INY
      CPY # $F0
      BNE UP
DOWN STX  $2005
      STY  $2005
      JSR YAN SHI
      DEY
      BNE DOWN
      BEQ UP
```

对应的 F BASIC 程序为：

```
5 REM No. 2－19b
10 A = 0
15 CLS : FOR I = 0 TO 600 : PRINT " A " ; : NEXT
20 POKE 8197 , 0
30 POKE 8197 , A
40 IF B = 0 THEN A = A + 1 : IF A = 240 THEN B = 1
50 IF B = 1 THEN A = A - 1 : IF A = 0 THEN B = 0
60 PAUSE 5
70 GOTO 20
```

四、画面的循环位移

画面的循环位移，包括画面的横向连续卷动、纵向连续滚动和斜向连续卷动三类位移。

1、画面的纵向连续滚动

所谓画面的纵向连续滚动就是画面的头尾相接，纵向旋转。画面纵向连滚的程序见程序 No. 2－20、No. 2－21。

(1) 画面向上连续滚动

No. 2－20a UP

```
    LDX # $00
    LDY # $00
UP  STX  $2005
    STY  $2005
```

```
    JSR YANSHI
    INY
    CPY #$F0
    BNE UP
    LDY #$00
    BEQ UP
```

对应的 F BASIC 程序如下：

```
5 REM No. 2-20b UP
10 A = 0
20 POKE 8197 , 0
30 POKE 8197 , A
40 A = A + 1 : IF A = 240 THEN A = 0
50 PAUSE 2
60 GOTO 20
```

(2) 画面向下连续滚动

No. 2-21a DOWN

```
     LDX #$00
     LDY #$EF
DOWN STX $2005
     STY $2005
     JSR YANSHI
     DEY
     BNE DOWN
     LDY #$EF
     BNE DOWN
```

对应的 F BASIC 程序为：

```
5 REM No. 2-21b DOWN
10 A = 239
20 POKE 8197 , 0
30 POKE 8197 , A
40 A = A - 1 : IF A = -1 THEN A = 239
50 PAUSE 2
60 GOTO 20
```

2、画面纵向连滚的工作原理

从程序 No. 2-20、No. 2-21的运行中可以明显看出，在画面的纵向连续滚动过程中，画面没有间断、没有跳跃，一直连续、匀速、圆滑地位移。这种现象，用图2-10的理论显然解释不通。那么，这种位移的工作原理是怎样的呢?

原来6527CPU 的显示页面并不象图2-10所示的那样简单地排列，实际上要复杂的多。为便于理解，我们采取分步讨论、循序过渡的方式进行解释。

几点假设：

(1) 把背景页假设为形如大口径水泥管子状的一个横向放置的圆筒，不妨称它为背景筒。显然，背景筒是由四个显示页面横向弯曲变形、上下两边对接而成的。

(2) 假设在背景筒外面又套上了一个稍大于背景筒的圆筒，不妨称它为显示筒。显示筒由不透光的材料制成，在显示筒上开有一个32列×30行或256×240点的矩形洞，这个矩形洞就是屏幕显示窗口。显然，现在的显示窗口已是一个曲面窗口。

(3) 假设在显示窗口的四个角上，向窗口的重心方向伸出四根支柱，以这四根支柱做支撑，在柱子的顶端搭一个平台，我们就站在这个平台上观察窗口。假定柱子足够长，我们离开显示窗口的距离足够远，则在我们的视觉中弯曲的窗口呈现出平面的图象。

以上三点假设如图2－11所示：

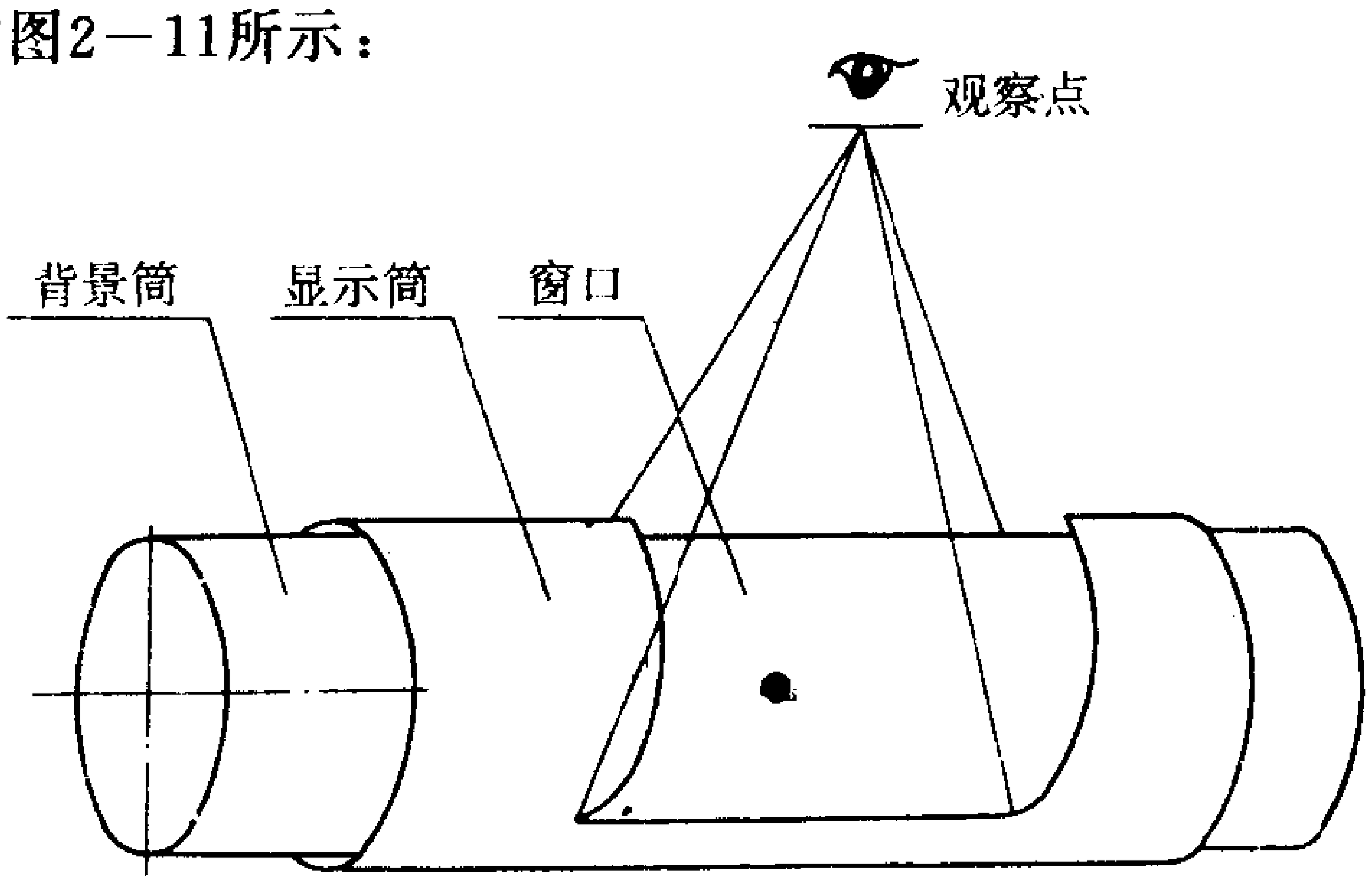


图2－11　画面纵向连滚原理示意图

(4) 假设显示筒由一同步电机带动，可绕背景筒正、反旋转；同时，还与一步进电机相连接，可带动显示筒沿背景筒左右滑动。

(5) 假设我们站在观察点上看屏幕，观察点随显示筒一起运动，观察者与显示窗口之间保持同步。

从以上五点假设出发，可很容易解释画面上下连滚时的工作原理。

·当画面在背景0页向上连滚时，则由程序 No. 2－20控制窗口在00、01两个显示页面连成的圆筒上面向下旋转。由于00、01两页显示同一幅画面，因此，在连续上滚时，看到的画面是上一行内容刚移出屏幕又从最底行显示出来。从视觉上似乎背景筒由一个页面组成，实际上，移出的是00页的第一行，下面显示的是01页的第一行。

·当画面在背景1页向上连滚时，则对＄2000单元的 D0位置1，相当于接通步进电机的电源，由步进电机把显示窗口推向1页背景筒部位。这时，窗口在由10、11显示页组成的背景筒上旋转，旋转显示过程同0页。

·画面向下连滚的情况与上滚基本相同，差别仅是令同步电机反转而已。

3、画面的横向连续卷动

画面的横向连续卷动就是00、10或01、11两幅显示页面头尾相连、连续左移或连续右移的过程。实现程序如下：

(1) 画面的连续左卷

No. 2－22a LEFT

```
  LDA  # $90
  STA  $2000
  STA  $FF
```

```
      LDX #$00
      LDY #$00
LEFT  STX $2005
      STY $2005
      JSR YANSHI
      INX
      BNE LEFT
      LDA $FF
      EOR #$01
      STA $FF
      STA $2000
      LDX #$00
      BEQ LEFT
```

对应的F BASIC程序如下：

```
5 REM No. 2—22b LEFT
10 A = 0
20 POKE 8197 , A
30 POKE 8197 , 0
40 A = A + 1 : IF A = 256 THEN 70
50 PAUSE 5
60 GOTO 20
70 IF B = 0 THEN POKE 8192 , 145 : B = 1 : GOTO 10
80 IF B = 1 THEN POKE 8192 , 144 : B = 0 : GOTO 10
90 END
```

运行程序No. 2—22以前，要先对背景0页、1页置入显示内容，以利观察

(2) 画面的连续右卷

画面的连续右卷是左卷的逆操作，只要修改No. 2—22b的10、40行语句就可实现。下面仅给出右卷的BASIC程序，见No. 2—23。

```
5 REM No. 2—23
10 A = 255
20 POKE 8197 , A
30 POKE 8197 , 0
40 A = A — 1 : IF A < 0 THEN 70
50 PAUSE 5
60 GOTO 20
70 IF B = 0 THEN POKE 8192 , 145 : B = 1 : GOTO 10
80 IF B = 1 THEN POKE 8192 , 144 : B = 0 : GOTO 10
90 END
```

4、画面横向连续卷动的工作原理

显然，图2—10b、图2—11都不能解释画面横向连卷的机理。我们可以采取画面纵向连滚的

假设思路，把画面设想为一个竖立的圆筒，相应的显示窗口也随之变动，示意见图2－12。

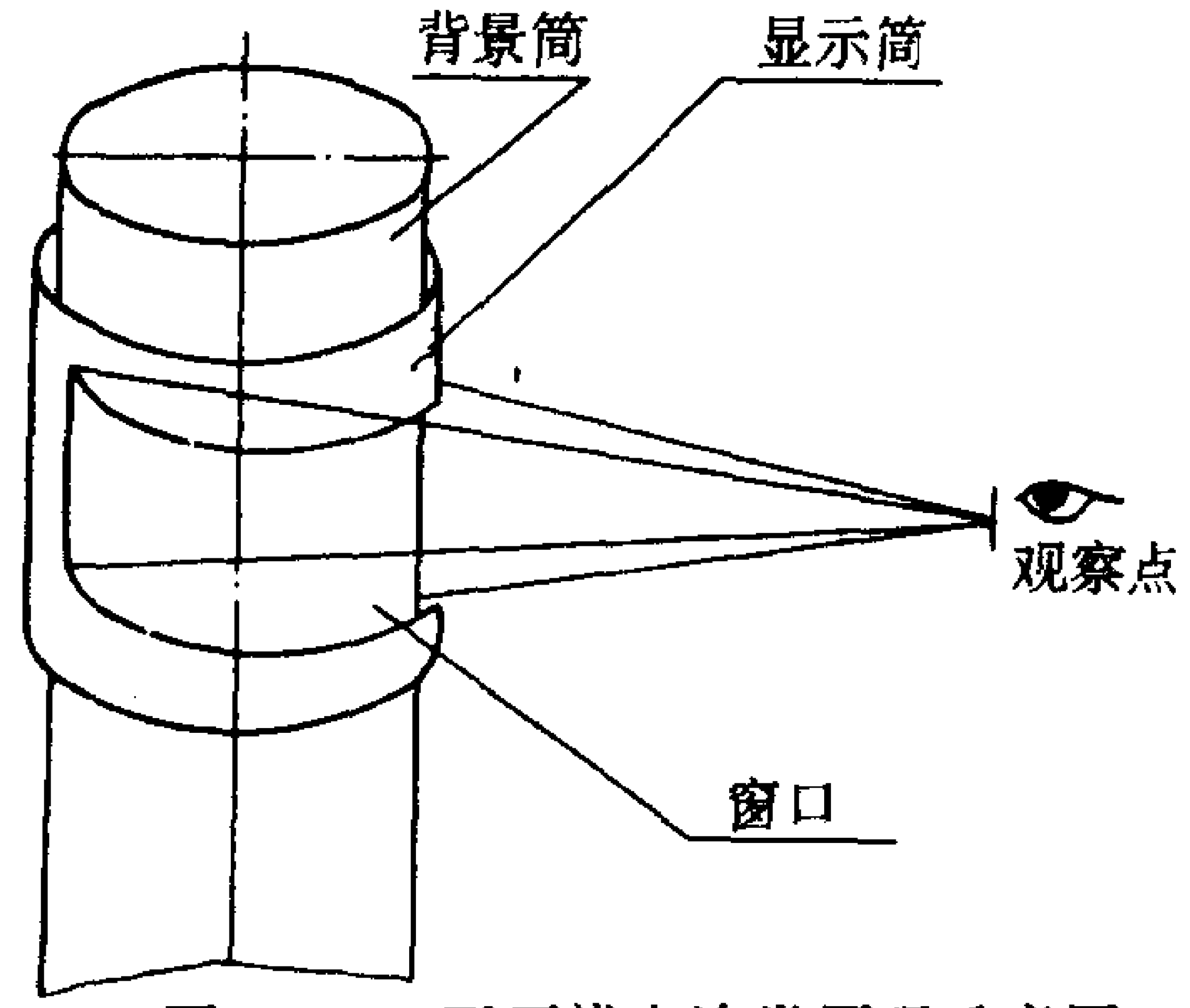


图2－12　画面横向连卷原理示意图

图1－12中，显示页面纵向弯曲，背景00、10和01、11页面的始端、末端对应相接，连成一竖立的圆筒，从而较好地解释了横向连卷的机理(具体过程留给读者分析。)

5、画面的斜向连续卷动的工作原理

把画面的纵向、横向连续位移程序合理搭配组合就可以方便地写出背景画面四个斜向连续卷动的程序。这一工作留给读者练习。下面重点讨论画面斜向连续卷动的工作机理。

当画面斜向连卷时，既有横向连卷的操作又有纵向连滚的过程。因此，图2－11、2－12都不能解释这一工作机理。那么，画面斜向卷动的工作机理应如何解释呢？

解决的方法是：对图2－11、2－12显示画面的变形过程综合考虑，把四页显示画面连接成为一个管状圆环。即把图2－11中的背景筒弯曲变形，两端口对接成环状，可称为背景环；相应地显示筒也弯曲变形为套在背景环上的一截圆环；显示窗口和观察点的设定仍如图2－11，这样就构成了图2－13的示意图。

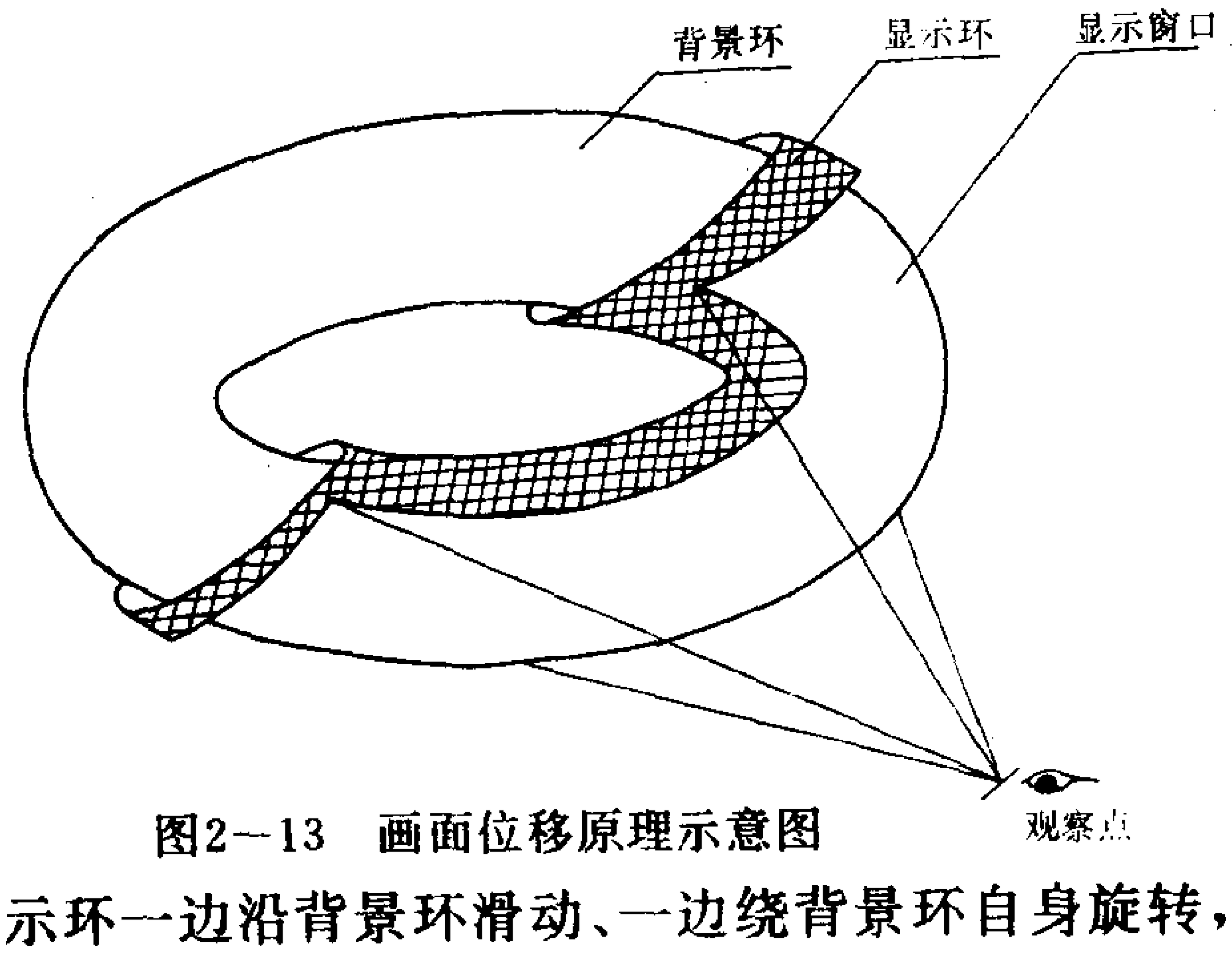


图2－13　画面位移原理示意图

在斜向连移过程中，显示环一边沿背景环滑动、一边绕背景环自身旋转，从而使四个显示页面的任一部分组合都能在窗口显示。

注——实物中，硬质的大环是不能绕小环旋转的。为了方便理解画面斜向连卷的工作机理，我们假定：显示环是由一种柔软的、富有弹性的材料制成，它既可沿背景环滑动又可绕背景环转动，且无论怎样运动，显示窗口都能与显示页面的形状相吻合。

根据以上假设，从图2－13出发，背景画面的各种位移工作原理都可得到圆满的解释：

·画面的横向往复位移过程是显示筒在背景筒上往复滑动；

·画面的横向连续位移是显示筒沿背景筒朝一个方向连续滑动；

·画面的纵向往复位移是显示筒绕背景筒往复旋转；

·画面的纵向连续位移是显示筒绕背景筒沿一个方向连续旋转；

·画面的斜向往复位移是显示筒一边绕背景筒往复旋转，一边沿背景筒往复滑动；

·画面的斜向连续位移是显示筒一边绕背景筒向一确定转向旋转，一边沿背景筒向一确定方向滑动。

以上讨论较全面地解释了引言中的第三类问题。

2·2·6 显示系统的配色

一、6527的配色区

我们知道，在 F BASIC 状态下，当使用 CGSET 指令选定背景和卡通配色板后，就可以在 DEF MOVE、DEF SPRITE 等指令中使用选定配色板中的配色代码。由于每个配色板有12种颜色，故两个配色板有24种颜色供选用。当我们欲自己组合颜色时，可使用 PALETB、PALETS 指令进行定义，一经定义后就可以在显示画面中使用。另外，我们还可以在背景配色板或卡通配色板的0＃配色代码的重新设定中改变底背景的配色，使用极为方便。但是，我们自己设定的配色数据存于系统 RAM 区的什么位置呢?查遍 RAM 区也找不到。原来6527CPU 工作系统有一专用的配色区，这一配色区位于 PPU 内部，寻址地址是＄3F00～＄3FFF。

二、6527配色区的地址分配

在 PPU 的＄3F00～＄3FFF 这256个单元中，存储了各种颜色代码，其地址分配见表2—8。

表2—8 PPU 配色地址分配表

地址	用途
3F00	底背景配色
3F01～3F03	背景配色板0＃配色代码的颜色代码
3F05～3F07	背景配色板1＃配色代码的颜色代码
3F09～3F0B	背景配色板2＃配色代码的颜色代码
3F0D～3F0F	背景配色板3＃配色代码的颜色代码
3F10	底背景配色
3F11～3F13	卡通配色板0＃配色代码的颜色代码
3F15～3F17	卡通配色板1＃配色代码的颜色代码
3F19～3F1B	卡通配色板2＃配色代码的颜色代码
3F1D～3F1F	卡通配色板3＃配色代码的颜色代码
*3F20～3FEF	映射于3F00～3F0F
*3F30～3FFF	映射于3F10～3F1F

注——①在＄3F20～＄3FEF 中凡以＄3F20、＄3F40、＄3F60等开始的连续的16 个单元均映射于＄3F00～3F0F。

②在＄3F30～＄3FFF 中凡以＄3F30、＄3F50、＄3F70等开始的连续的16 个单元均映射于＄3F10～3F1F。

③凡＄3F04、＄3F08、＄3F0C、＄3F14、＄3F18、＄3F1C 等最末一位是4、8、CH 的单元没有使用，一般置入0FH。

例2—4 写出在 F BASIC 状态下执行“CGSET 1,0”和“CGSET 0,2”命令后各配色单元的内容。

执行“CGSET 1,0”后，使用读 PPU 内存的程序可读出各配色单元内容如下：

3F00	0F	30	21	02	0F	30	27	18
3F08	0F	30	27	16	0F	29	36	17
3F10	0F	36	16	02	0F	27	30	19
3F18	0F	35	25	17	0F	30	27	16

执行"CGSET 0,2"后，各配色单元的内容如下：

3F00	0F	2C	15	07	0F	27	21	12
3F08	0F	29	36	17	0F	30	26	07
3F10	0F	30	26	12	0F	30	15	12
3F18	0F	30	12	16	0F	30	26	19

当我们执行

"PALETB 0 , 3 , 5 , 7 , 9"

时则底背景变为蓝色，相应的配色单元变为

3F00 03 05 07 09 0F 27 21 12

读者可使用读 PPU 内存的方法查看、验证各配色单元的内容；也可以用直接写入 PPU 的方法，不使用 PALET 指令而改变底背景或背景、卡通的配色。

三、说明

1、在配色区，每四个单元为一组，每组的后三个单元存颜色代码。

2、在各个配色单元中，只有 PPU 地址的末位为0的单元为底背景配色单元，其它位无效。

3、如果把＄3F00～＄3FFF 分成16行，那么偶数行之间互相等效(包括0行，即0、2、4、6……)，也就是说对＄3F60～＄3F6F 操作等效于对＄3F00～3F0F 操作，都作用于背景页；同样奇数行之间互相等效(1、3、5、7……)，它们都作用于卡通页。

2·2·7 图形库结构与卡通设计

一、图形库结构

在任天堂游戏中，无论背景、卡通都采用了点阵块显示的方法。采用这种方法的优点是：画面处理速度快、占用系统内存少。

1、图形库在内存中的位置

图形库占用 PPU 所管理内存的＄0000～＄1FFF 的8K 空间。一般使用8×8K 的 EPROM2764存放，它共分两个区：第Ⅰ区、第Ⅱ区。

第Ⅰ区占用＄0000～＄0FFF 4K 空间，存放256个8×8点阵的卡通图形块(包括字符)。

第Ⅱ区占用＄1000～＄1FFF 4K 空间，存放256个8×8点阵的背景图形块(包括字符)。

我们可使用程序 No. 2－6或 No. 2－7读出图形库数据(见附录)以进行研究。

2、图形结构

在图形库中，从＄0000开始，每连续的16个字节为一段，定义一个8×8点阵的图形块。实际上，每一个8×8点阵的图形是由2个8×8点阵的子图形叠加而成的。结构如下：

(1) 把16个字节分成两部分，前8个字节为第一部分、后8个字节为第二部分。这两部分分别对应于一个8×8点阵的子图形。

(2) 把两个子图形叠加在一起，构成一个8×8点阵的图形块。这样，每个点就由两位二进制数表示，从而可使用四种颜色组出各种各样的图形块。

下面通过实例说明图形块的定义过程。

例2—5　定义一个图形块，要求：用0＃配色为底色；用1＃配色围成一个方框，在方框内有一个罗马数字“Ⅰ”；要求用2＃配色写横；3＃配色写竖。图形块如图2—14所示。

定义过程如下：

(1) 画一个8×8的方格，如图2—15。

(2) 把最外圈的小方格中填入1＃配色代码“02”，使其构成1＃配色的方框。填入数字时，要求一个方格中的数字分两行填写。填入“01”时，上一行填“0”、下一行填“1”。(下同)

(3) 从第三列开始填入四个“10”，构成“Ⅰ”的上横。从第三列第六行开始填入四个“10”，以构成“Ⅰ”的下横。

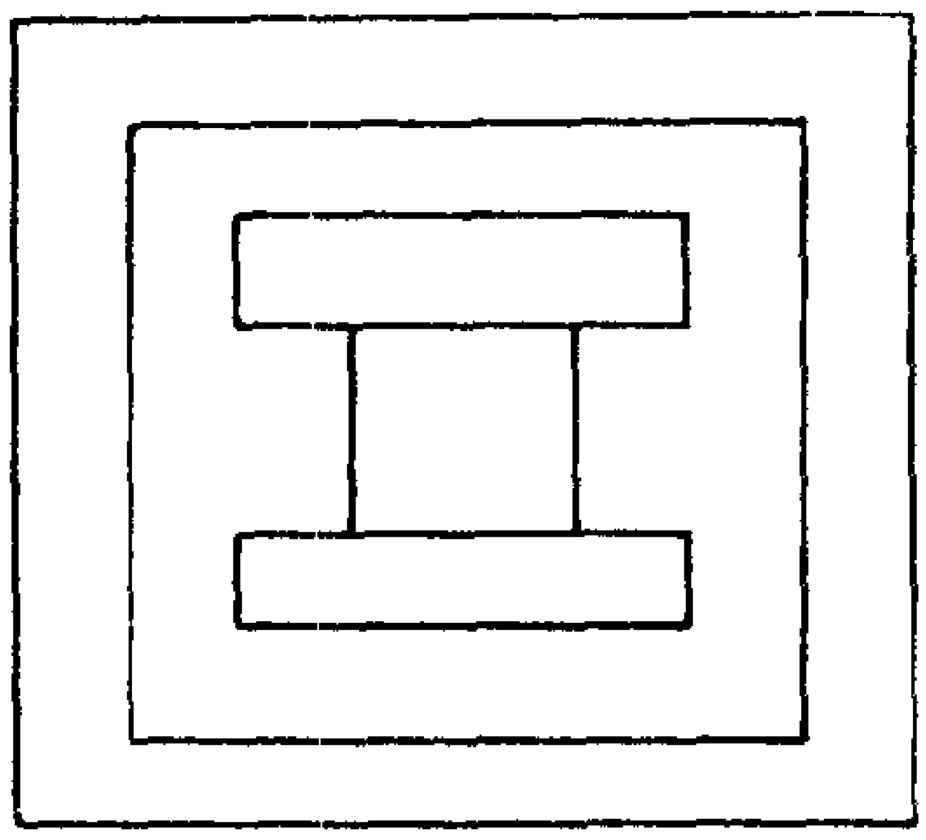


图2—14

行：	列：	1	2	3	4	5	6	7	8	
一	00	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	FF
二	00	0 1	0 0	0 0	0 0	0 0	0 0	0 0	0 1	81
三	3C	0 1	0 0	1 0	1 0	1 0	1 0	0 0	0 1	81
四	18	0 1	0 0	0 0	1 1	1 1	0 0	0 0	0 1	99
五	18	0 1	0 0	0 0	1 1	1 1	0 0	0 0	0 1	99
六	3C	0 1	0 0	1 0	1 0	1 0	1 0	0 0	0 1	81
七	00	0 1	0 0	0 0	0 0	0 0	0 0	0 0	0 1	81
八	00	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	FF

图2—15

从第四列第四行开始填入两个“11”，从第四列第五行开始填入两个“11”，以构成“Ⅰ”的一竖。

(5) 把空白方格填入“00”。

(6) 把每一格中上行的8位二进制数写在方框的左边；把下行的8位二进制数写在方框的右边；

(7) 先自上而下抄下左边的8个数据，再抄右边的8个数据。这16个数据就是例2—5的图形数据。它们为：

00　00　3C　18　18　3C　00　00

FF　81　81　99　99　81　81　FF

若以空白“□”代表0＃配色、以“■”代表1＃配色、以“※”代表2＃配色、以“●”代表3＃配色，对图2—15进行涂色后，其效果如图2—16所示。

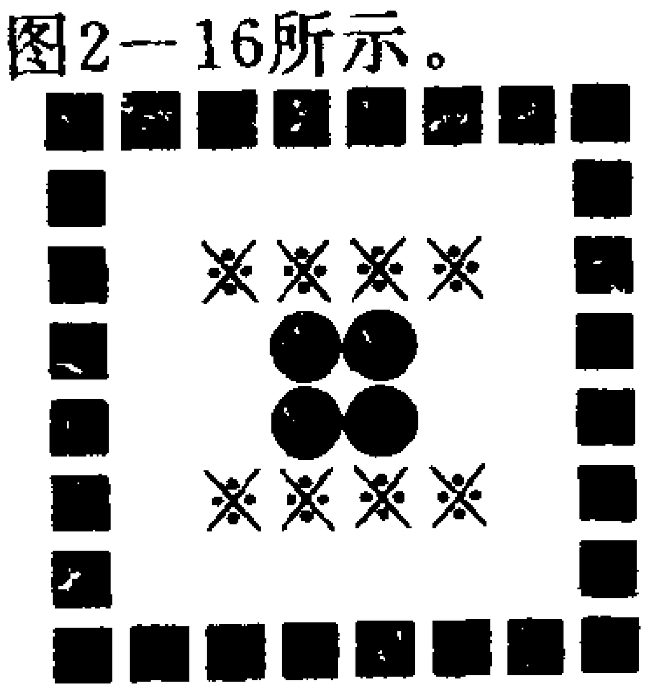


图2—16

二、卡通图形的设计

卡通图形的设计过程与背景图形大同小异，所不同的是：卡通图形一般都是16×16点阵的。一个卡通图形要由四个8×8点阵的卡通块组成。因此，在定义卡通图形时，要把一个卡通图形分为四份，每一个卡通块占图形的四分之一。我们先验证一下 F BASIC 中马里欧哥哥的定义数据。

4

00	0 0	0 0	0 0	0 0	0 0	0 0	0 1	0 1	03
00	0 0	0 0	0 0	0 0	0 1	0 1	0 1	0 1	0F
07	0 0	0 0	0 0	0 0	0 0	1 0	1 0	1 0	00
1F	0 0	0 0	0 0	1 0	1 0	1 0	1 1	1 0	02
3F	0 0	0 0	1 0	1 0	1 0	1 1	1 0	1 0	04
1F	0 0	0 0	0 0	1 1	1 1	1 1	1 1	1 0	1E
0F	0 0	0 0	0 0	0 0	1 0	1 0	1 0	1 0	00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 1	0 1	03

5

00	0 1	0 1	0 0	0 0	0 0	0 0	0 0	0 0	C0
00	0 1	0 1	0 1	0 0	0 0	0 0	0 0	0 0	E0
E0	1 1	1 1	1 1	0 0	0 0	0 0	0 0	0 0	E0
F0	1 0	1 1	1 0	1 1	0 0	0 0	0 0	0 0	50
F0	1 1	1 1	1 0	1 1	0 0	0 0	0 0	0 0	D0
F0	1 0	1 0	1 1	1 1	0 0	0 0	0 0	0 0	30
C0	1 0	1 0	0 0	0 0	0 0	0 0	0 0	0 0	00
80	1 1	0 1	0 1	0 0	0 0	0 0	0 0	0 0	E0

6

01	0 0	0 0	0 0	0 0	0 0	0 1	0 1	1 1	07
11	0 0	0 0	0 0	1 0	0 1	0 1	0 1	1 1	0F
1F	0 0	0 0	0 0	1 0	1 1	1 1	1 1	1 1	0F
0F	0 0	0 0	0 0	0 0	1 1	1 1	1 1	1 0	0E
07	0 0	0 0	0 0	0 0	0 0	1 1	1 1	1 0	06
01	0 0	0 0	0 0	0 0	0 0	0 1	0 1	1 1	07
00	0 0	0 0	0 0	0 0	0 1	0 1	0 1	0 0	0E
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 1	01

7

00	0 1	0 1	0 1	0 0	0 0	0 0	0 0	0 0	E0
80	1 1	0 1	0 1	0 0	0 0	0 0	0 0	0 0	E0
80	1 0	0 1	0 1	0 0	0 0	0 0	0 0	0 0	60
A0	1 0	0 1	1 1	0 0	0 0	0 0	0 0	0 0	60
E0	1 0	1 1	1 1	0 0	0 0	0 0	0 0	0 0	60
C0	1 1	1 1	0 0	0 0	0 0	0 0	0 0	0 0	C0
00	0 1	0 1	0 0	0 0	0 0	0 0	0 0	0 0	C0
00	0 1	0 1	0 0	0 0	0 0	0 0	0 0	0 0	C0

图2--17 玛丽走2的图形数据

查阅附录中的卡通图形库，＄0040～＄007F 连续的64个单元是马里欧走2的点阵数据，现抄录如下：

```
0040  00  00  07  1F  3F  1F  0F  00
      03  0F  00  02  04  1E  00  03
      00  00  E0  F0  F0  F0  C0  80
      C0  E0  E0  50  D0  30  00  E0
      01  11  1F  0F  07  01  00  00
      07  0F  0F  0E  06  07  0E  01
      00  80  80  A0  E0  C0  00  00
      E0  E0  60  60  60  C0  C0  C0
```

把数据填入4个8×8的方格表中。见图2－17。

若仍按例2－5配色代码所代表色块的约定，对图2－17涂色，将得到马里欧走2的图形，见图2－18。

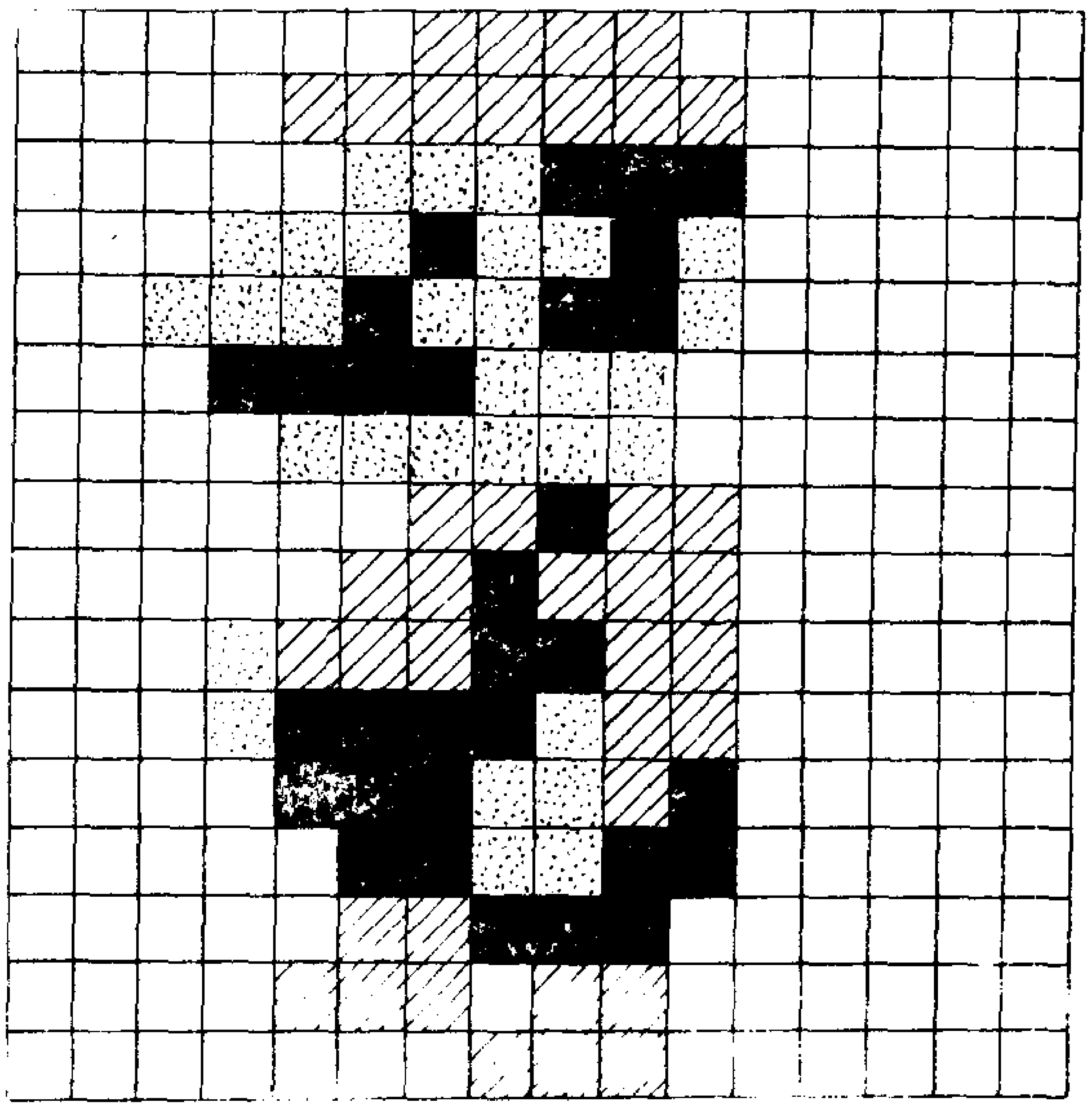


图2－18　马里欧走2图形

凡玩过《超级马里欧》游戏的朋友都知道，游戏开始时马里欧只有16×16点阵大小，当吃到红蘑菇后，身体立即长大为32×32点阵大小。但F BASIC中没有这么大的卡通图形，它的图形数据将是怎样的呢？这实际上并不神秘，它不过是把马里欧的16×16点阵的图形放大一倍而已。下面就讨论32×32点阵马里欧图形的定义方法。

例2－6　试求32×32点阵的马里欧走2的图形数据。

现在，我们已经掌握了16×16点阵的马里欧走2的图形数据，那么求32×32点阵的图形数据将变的十分容易。凡实践过图形的方格放大的朋友都清楚：当要把图形放大一倍时，只要把方格的尺寸比原图方格扩大一倍，然后照原图填入每个方格中的内容，就得到了放大了的图象。因此，对例2－16，只要把原16×16点阵图形中的4个8×8点阵图形块分成16个4×4点阵块；然后把每一个4×4点阵块的数据填入8×8点阵中，即4×4点阵中的一个方格中的数据要填入8×8点阵中4个相邻的方格中；最后把这16个8×8点阵的图形块组合在一起，就完成了32×32点阵马里欧走2的图形设计。

下面我们仅以原图形库中马里欧走2的第一个8×8点阵块放大为例，介绍图形放大方法。

1、读出马里欧走2的第一个图形块数据（即图形库第Ⅰ区，序号为4的图形数据）。从附录中可查出这16个数据为：

0040　00　00　07　1F　3F　1F　0F　00

0048　03　0F　00　02　04　1E　00　03

2、把以上16个数据分成4组，每组4个，分别排列如下：

(1) 00　00　07　1F

(2) 03　0F　00　02

(3) 3F　1F　0F　00

(4) 04　1E　00　03

3、把以上4组数据中的每一个数据按高、低位拆开，高位为一组、低位为一组，共组成八组。排列如下：

A ┌ (1) 0　0　0　1
　└ (2) 0　0　0　0

B ┌ (1)' 0　0　7　F
　└ (2)' 3　F　0　2

C (3) 3 1 0 0　　D (3)' F F F 0
  (4) 0 1 0 0　　 (4)' 4 E 0 3

4、把每一组数字按表2－9的规则替换成相应的数据。

表2－9　图形放大数据换算表

原始数据	替换数据	原始数据	替换数据
0	00	8	C0
1	03	9	C3
2	0C	A	CC
3	0F	B	CF
4	30	C	F0
5	33	D	F3
6	3C	E	FC
7	3F	F	FF

以上各组替换完毕后的数据如下：

A (1) 00 00 00 03
  (2) 00 00 00 00

B (1)' 00 00 3F FF
  (2)' 0F FF 00 0C

C (3) 0F 03 00 03
  (4) 00 03 00 00

A (1)' FF FF FF 03
  (2)' 30 FC 00 0F

5、把以上各组中的数据每一个重写一次，使每行由4个数据变为8个数据。则有：

A (1) 00 00 00 00 00 00 03 03
  (2) 00 00 00 00 00 00 00 00

B (1)' 00 00 00 00 3F 3F FF FF
  (2)' 0F 0F FF FF 00 00 0C 0C

C (3) 0F 0F 03 03 00 00 03 03
  (4) 00 00 03 03 00 00 00 00

D (1)' FF FF FF FF FF FF 00 00
  (2)' 30 30 FC FC 00 00 0F 0F

以上A、B、C、D四组数据，即为4号卡通块放大一倍后的图形数据。现在，我们把它写入方格验证一下。

验证情况见图2－19、图2－20。

00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00
03	0 0	0 0	0 0	0 0	0 0	0 0	1 0	1 0	00
03	0 0	0 0	0 0	0 0	0 0	0 0	1 0	1 0	00

00	0 0	0 0	0 0	0 0	0 1	0 1	0 1	0 1	0F
00	0 0	0 0	0 0	0 0	0 1	0 1	0 1	0 1	0F
00	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	FF
00	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	FF
3F	0 0	0 0	1 0	1 0	1 0	1 0	1 0	1 0	00
3F	0 0	0 0	1 0	1 0	1 0	1 0	1 0	1 0	00
FF	1 0	1 0	1 0	1 0	1 1	1 1	1 0	1 0	0C
FF	1 0	1 0	1 0	1 0	1 1	1 1	1 0	1 0	0C

0F	0 0	0 0	0 0	1 0	1 0	1 0	1 0	1 0	00
0F	0 0	0 0	0 0	0 0	1 0	1 0	1 0	1 0	00
03	0 0	0 0	0 0	0 0	0 0	0 0	1 1	1 1	03
03	0 0	0 0	0 0	0 0	0 0	0 0	1 1	1 1	03
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00
00	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	00

FF	1 0	1 0	1 1	1 1	1 0	1 0	1 0	1 0	30
FF	1 0	1 0	1 1	1 1	1 0	1 0	1 0	1 0	30
FF	1 1	1 0	1 1	1 1	1 1	1 1	1 0	1 0	FC
FF	1 1	1 1	1 1	1 1	1 1	1 1	1 0	1 0	FC
FF	1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0	00
FF	1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0	00
00	0 0	0 0	0 0	0 0	0 1	0 1	0 1	0 1	0F
00	0 0	0 0	0 0	0 0	0 0	0 1	0 1	0 1	0F

图2—19　4＃卡通(马里欧走2)放大一倍后的图形数据

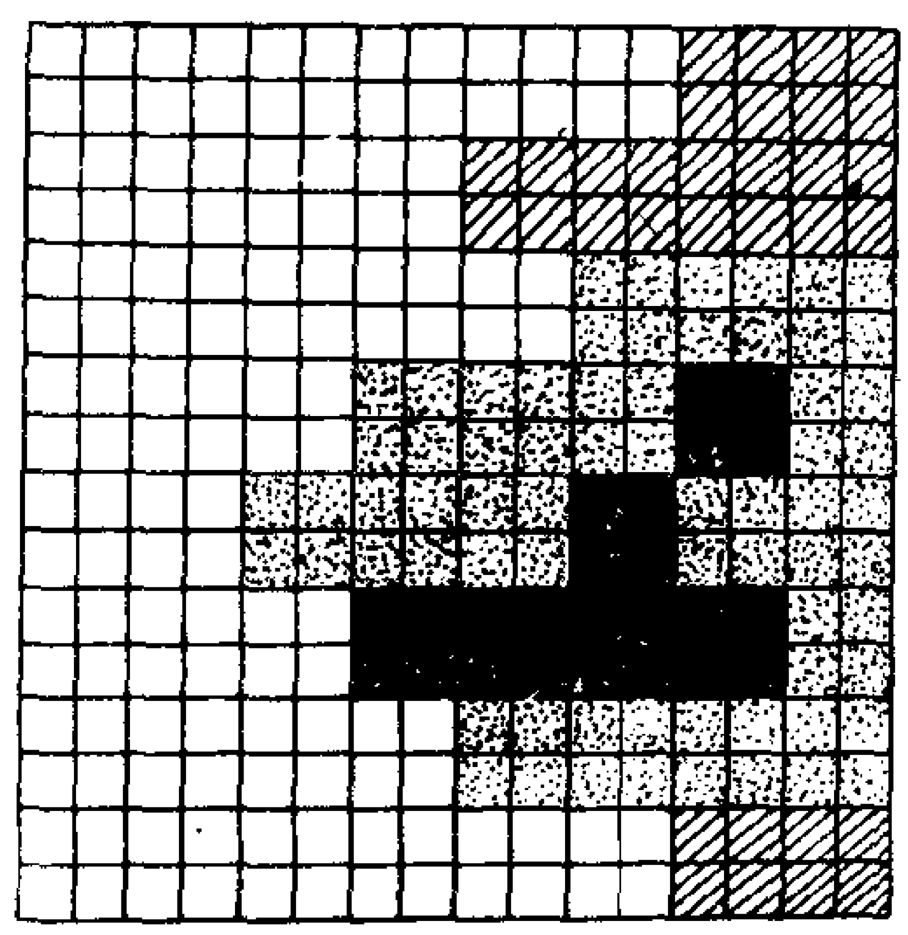


图2—20　4＃卡通(马里欧走2)放大一倍后的图形

可见，放大的图形数据都是正确的。

仿照以上五个步骤，可以写出后三个卡通块5＃、6＃、7＃的放大数据。有兴趣的读者可以自己做一遍。下面仅给出最后的结果(以 A、B、C……为序)：

E	00	00	00	00	FC	FC	FF	FF
	F0	F0	FC	FC	FC	FC	33	33
F	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00
G	FF	FF	FF	FF	F0	F0	C0	C0
	F3	F3	0F	0F	00	00	FC	FC
H	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00
I	00	00	03	03	03	03	00	00
	00	00	00	00	00	00	00	00
J	03	03	03	03	FF	FF	FF	FF
	3F	3F	FF	FF	FF	FF	FC	FC
K	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00
L	3F	3F	03	03	00	00	00	00
	3C	3C	3F	3F	FC	FC	03	03
M	00	00	C0	C0	C0	C0	CC	CC
	FC	FC	FC	FC	3C	3C	3C	3C
N	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00
O	FC	FC	F0	F0	00	00	00	00
	3C	3C	F0	F0	F0	F0	F0	F0
P	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00

读者可把以上12组数据填入12个8×8方格中，以对比放大后的效果。

通过以上讨论，我们基本弄通了图形块设计的方法步骤。同时也解释了我们的电脑游戏机中为什么没有《魂斗罗》、《双截龙》等游戏中的背景画面和卡通的缘故。明白了这一道理后，我们可以随意设计自己喜欢、需要的图形库图形数据，然后通过 EPROM 写入器写入芯片，替换出学习卡中的字模存储器。这样就可以在编程中使用自己设计的图形了。

当然，在实际的图形设计中，应该编写一个工具软件，让电脑把设计好的图形自动转换成图形数据或者截取、借鉴其它游戏软件的图形数据，以提高图形设计效率。这些问题已超出本书的讨论范围，恕不赘述。

2·3 6527CPU 的音响系统

任天堂游戏中美妙动听的音响来自6527CPU 内部的 PSG（Programmable Sound Generator）——可编程音响发生器。其主要特点为：

·五个声部发声：三个主声部、两个副声部；

·15级音量调节；

·15种音色选择；

·15种音形包络；

·8种包络周期；

·4096种音调。

以上六大特点表明了6527CPU 音响系统的发音功能是十分强大的。只要对发声软开关设

置恰当的数值就可以模拟出各种各样的音响。众所周知,《魂斗罗》二代游戏专门设置了音乐欣赏密码开关,只要选中密码便可欣赏游戏中的全部音响。实际上,这些音响我们都可以在电脑游戏机中通过设置发声软开关模仿出来。

强大的发音功能、简单的发音操作是6527CPU 优于其它 CPU 的又一显著特点。

2·3·1 发声软开关简介

6527CPU 的音响发声软开关位于 CPU I/O 区的＄4000～＄4013共20个单元,由它们组成五个发声声部,每个声部占用4个单元。其分配顺序如下:

·第一主声部:＄4000～＄4003

·第二主声部:＄4004～＄4007

·第三主声部:＄4008～＄400B

·第一副声部:＄400C～＄400F

·第二副声部:＄4010～＄4013

各声部对应单元的功能相同。下面仅以第一主声部为例介绍各软开关的作用。

1、音色音量控制软开关＄4000

＄4000是第一主声部的音色、音量控制软开关,各位功能如图2－22:

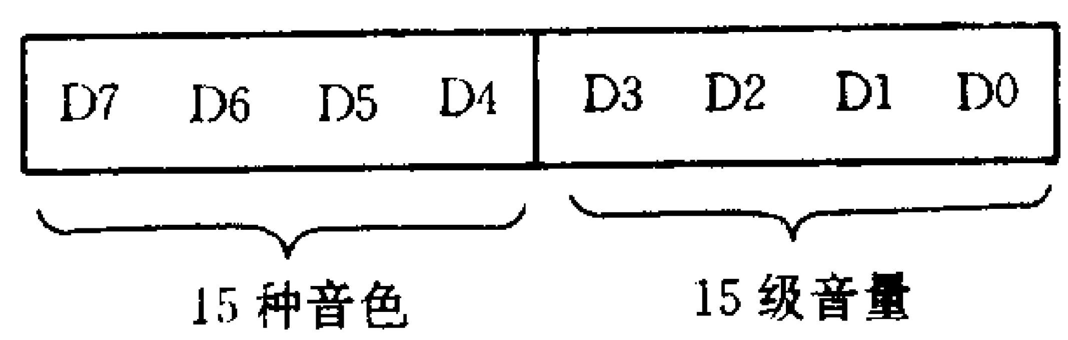


图2－22

·D0～D3:为15级音量控制软开关。其值为0时关闭音量;为1时音量最小;为15时音量最大。

·D4～D7:为音色控制软开关。其中 D4＝1时发声时间延长;D5＝1时可发出各段衰减音;D6＝1时将缩短发声时间。

2、包络控制软开关＄4001

＄4001是第一主声部音形包络控制软开关。各位功能如图2－23。

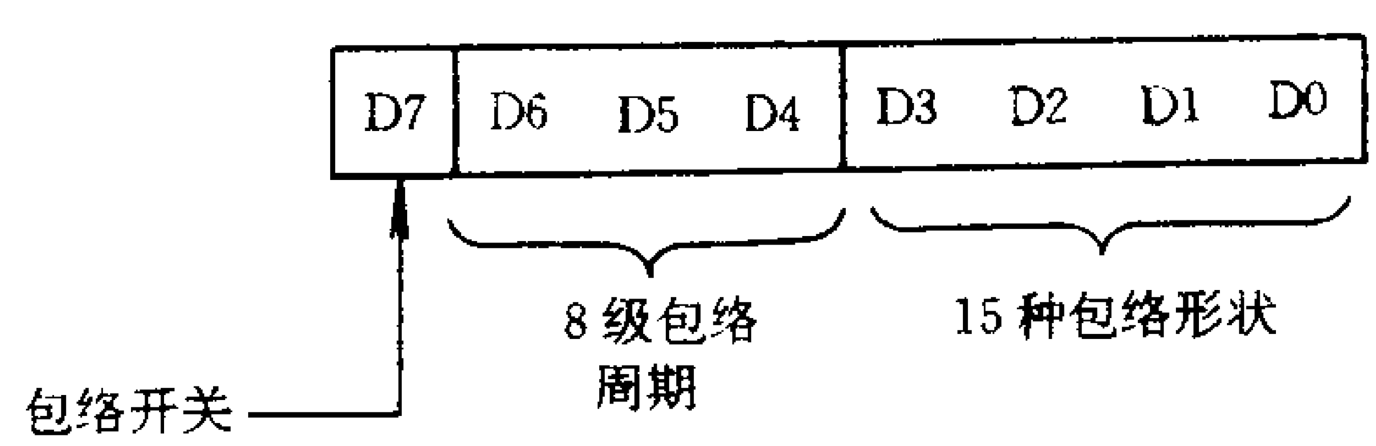


图2－23

·D7:为包络开关位。D7＝1时包络有效。

·D4～D6:为8级包络周期控制开关。调整它们的取值,可模拟声差极大的效果音响。它们与＄4000的 D5位配合可发出多级衰减音。

·D0～D3：为15种包络形状控制开关，取值范围0～15。当取值等于0和等于8时无包络，发平直音；当取值等于1～7时，发音频率从高到低变化；当取值等于9～5时，发声频率则从低到高。当取值等于1、9时发声时间最短；取值等于7、15时发声时间最长。

3、音调细调控制软开关＄4002

＄4002是第一主声部音调细调控制软开关。使用它可以细调发声音调，取值范围0～255，取值越大发声频率越低。

4、音调粗调控制软开关＄4003

＄4003是第一主声部音调粗调开关。使用它可以粗调发声音调，取值范围0～15，取值越大频率越低。

5、发声开关组＄4015

＄4015是启动发声控制开关组，写入不同的数据，可选择不同的发声声部。各位控制功能见图2－24。

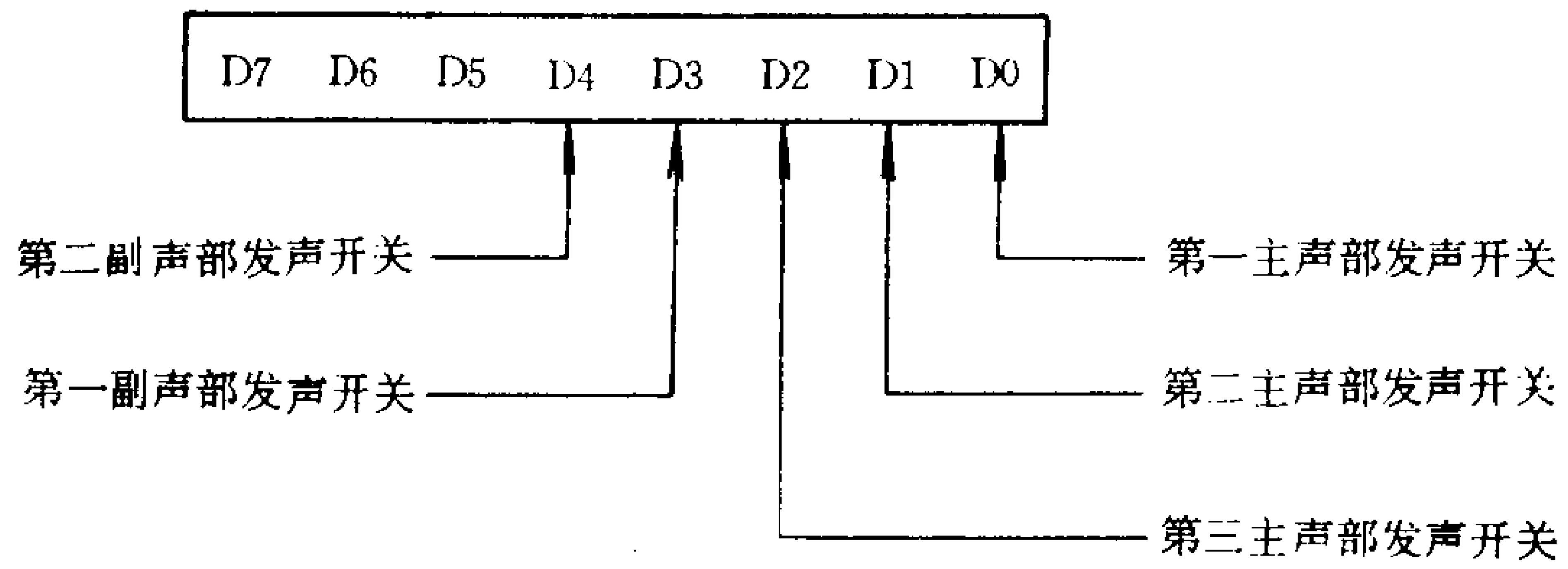


图2－24

根据图2－24，向不同位置值可方便地控制各声部发声。置入1接通开关；置入0则关闭开关。

2·3·2 发声方法及控制程序

一、发声方法

6527CPU的发声方法就是对不同的声部单元置入不同的数值。置数顺序一般为：

1、接通发声开关，做好发声准备。一般对＄4015单元置入FFH，接通五个声部的发声开关。若不需要哪一声部发声时，可把其对应位置入0。

2、对包络开关单元、音调细调单元、音调粗调开关单元置入数据，以确定发声音调。

3、最后对音色、音量控制开关单元置入音色音量数据，就可启动发声。这时，程序可转向处理其它事务，只要不对＄4015写入0或对发声声部的音量单元写入0，发声可一直持续下去。

二、发声控制程序

1、控制发声的原理程序

控制发声的机器语言程序见No. 2－24。

No. 2－24

```
LDA # $FF
STA $4015     ;打开发声开关，准备发声
LDA # $b1
STA $4001     ;第一主声部包络数据
```

```
LDA  # $c1
STA  $4002      ;第一主声部音调细调
LDA  # $d1
STA  $4003      ;第一主声部音调粗调
LDA  # $a1
STA  $4000      ;第一主声部音色音量
LDA  # $b2
STA  $4005
LDA  # $c2
STA  $4006
LDA  # $d2
STA  $4007
LDA  # $a2
STA  $4004      ;第二主声部数据
LDA  # $b3      ;以后各声部数据
……
RTS
```

2、F BASIC 发声程序

下面仅以第一主声部为例介绍在 F BASIC 状态下的发声程序，见 No. 2—26。

```
5 REM No. 2—26
10 POKE &H4015 , 255
20 POKE &H4000 , a
30 POKE &H4001 , b
40 POKE &H4002 , c
50 POKE &H4003 , d
60 A $ =INKEY $ (0) : GOTO 10
```

输入程序 No. 2—26，按表2—10填入 a、b、c、d 发声参数，就可模拟出不同的效果音。

表2—10　第一主声部效果音发声数据

效果音响	a	b	c	d	效果音响	a	b	c	d
被枪击中	1F	81	EF	08	八级升调衰减音	2F	DF	EF	08
飞机降落	1F	96	EF	08	十级升调衰减音	2F	EF	EF	08
子弹射到金属上	1F	89	EF	08	十一级升调衰减	2F	FF	EF	08
射击声	1F	8B	EF	08	长升调音	3F	BF	EF	08
七宝奇谋进小屋	1F	FA	EF	08	三级降调衰减音	AF	F4	EF	08
警报	FF	FF	FF	FF	五级降调衰减音	AF	F5	EF	08
铃声	AF	C0	EF	08	十级降调衰减音	AF	F6	EF	08
四级升调衰减音	2F	BE	EF	08	廿级降调衰减音	AF	F7	EF	08
六级升调衰减音	2F	BF	EF	08	合卡箭头移动声	1F	99	EF	08
七级升调衰减音	2F	CF	EF	08	合卡节目换页声	1F	AA	EF	08

3、火车运行声音模拟

使用第一副声部可以维妙维肖、十分逼真地模拟出火车运行中的各种声响，其中有火车上

坡、下坡的加速、减速声、火车钻入山洞的回声、火车过桥梁时的隆隆声，还有火车走远的渐弱声和火车开近的渐强声等。程序 No. 2－27仅仅数行程序就能产生如此奇妙的效果，这不能不令人对6527CPU 的发声能力惊叹咋舌。下面列出程序清单供朋友们欣赏。

```
5 REM No. 2－27
10 B = 33
20 POKE &H4015 , 8
30 POKE &H400F , &H80
40 POKE &H400C , B
50 POKE &H400D , &HFF
60 POKE &H400E , A
70 PAUSE RND ( 100 ) + 150 : A = A + 1
80 IF A = 250 THEN A = RND ( 30 ) + 10
90 IF A > 200 THEN B = B + 1
100 IF B > 47 THEN B = RND ( 5 ) + 33
110 GOTO 30
```

4、音乐旋律的演奏

以上仅介绍了各种效果音的发声方法，下面讨论音乐旋律的演奏。

6527CPU 可发出6个8度音域共72个音阶（包括半音）。因此，几乎可以演奏常见的各种音乐旋律。

演奏时，当音色选定后，音符的变化就是改变音调单元的数据。音符与音调的对应关系见表2－11。

关于音符时值的控制（发声音长），我们不必像苹果机、中华学习机那样采用延时的方法进行控制（那样不仅降低了 CPU 的工作效率，而且也无法做到声、像同步），而是通过设置计数器的方法实现的，一般放在中断程序中进行处理。由于6527CPU 的中断频率为50HZ，中断间隔为0,02秒，因此使用中断次数控制音长，可准确地控制音符的时值。在实际演奏中，音长、音量、音调各数值都是通过查表法获得的。这些内容我们将在以后章节中讨论。

5、各声部的特点及应用

发声系统的五个声部各有特点，使用方法也不尽相同。一般用第一声部作为主音演奏，用第二主声部作为和声伴奏，或两个声部交换位置。也就是说第一、第二主声部具有近似相同的特性。第三主声部主要用于和声伴奏，即使对其置入同第一、第二主声部相同的数据，第三主声部的发声音调也要比它们低一个八度音，同时音符的时值也缩短很多。这样安排的目的是节约数据区。两个副声部主要用于模仿各种噪声，如鼓声、锣声、钟声、脚步声、蝉鸣声等效果音。火车运行声的模拟就是使用第一副声部实现的。另外，各声部之间还有着密切的内在联系，只要合理搭配，就能够发出千奇百怪的奇妙音响。有兴趣的读者不妨耐心一试。

以上讨论，我们大体上解释了引言中关于音响的问题。至于具体的音乐演奏程序的编写和背景音乐的插入方法，将在第五章详细介绍。

表2－11　音符与音调数据对照表

音符	音调细调	音调粗调	音符	音调细调	音调粗调	音符	音调细调	音调粗调
1̤	AE	0E	2	7C	09	3̈	53	08
1̤#	4E	0E	2#	67	09	(3#)4̈	4F	08
2̤	F3	0D	3	52	09	4̈#	4A	08
2̤#	9E	0D	(3#)4	3F	09	5̈	46	08
3̤	4D	0D	4#	2D	09	5̈#	42	08
(3#)4̤	01	0D	5	1C	09	6̈	3E	08
4̤#	B9	0C	5#	0C	09	6̈#	3A	08
5̤	75	0C	6	FD	08	7̈	37	08
5̤#	35	0C	6#	EE	08	(7#)1⃛	34	08
6̤	F8	0B	7	E1	08	1⃛#	31	08
6̤#	BF	0B	(7#)1̇	D4	08	2⃛	2E	08
7̤	89	0B	1̇#	C8	08	2⃛#	2B	08
(7#)1̣	57	0B	2̇	BD	08	3⃛	29	08
1̣#	27	0B	2̇#	B2	08	(3#)4⃛	27	08
2̣	F9	0A	3̇	A8	08	4⃛#	24	08
2̣#	CF	0A	(3#)4̇	9F	08	5⃛	22	08
3̣	A6	0A	4̇#	96	08	5⃛#	20	08
(3#)4̣	80	0A	5̇	8D	08	6⃛	1E	08
4̣#	5C	0A	5̇#	85	08	6⃛#	1C	08
5̣	3A	0A	6̇	7E	08	7⃛	1B	08
5̣#	1A	0A	6̇#	76	08			
6̣	FC	09	7̇	70	08			
6̣#	DF	09	(7#)1̈	69	08			
7̣	C4	09	1̈#	63	08			
(7#)1	AB	09	2̈	5E	08			
1#	93	09	2̈#	58	08			

2·4 6527CPU 的中断系统

我们知道,6527CPU 最常用的中断方式是非屏蔽中断 NMI。所谓非屏蔽中断是指不能通过编程进行阻止的中断。只要非屏蔽中断请求信号一发出,中央处理器在执行完当前指令后就立即响应。

1、NMI 中断的执行过程

6527CPU 的中断入口地址放在＄FFFA(地址低位)和＄FFFB(地址高位)中,F BASIC 中一般指向＄00ED。

当 CPU 响应中断时,则自动保存断点地址和状态,即将当前的程序计数器 PC 值和状态寄存器 P 值送入堆栈保存;然后根据中断矢量单元＄FFFA、＄FFFB 提供的中断程序入口地址跳转中断处理程序进行中断处理;当中断处理结束后,则返回到断点地址处,恢复原来的内部状态,继续执行原程序。

2、NMI 中断的作用

NMI 中断在6527CPU 的工作系统中起着巨大的作用。任天堂游戏中的音响发出、画面更换、角色控制等操作都是在中断程序中完成的;人机对话工作系统的全部操作,几乎都是在中断中进行的;F BASIC 工作系统中,动作卡通的运动也是在中断中实现的。因此,NMI 中断在6527CPU 工作系统中占有举足轻重的地位。甚至可以这样说:没有 NMI 中断就没有任天堂游戏。强大的中断功能是6527CPU 的又一重要特点。

3、中断程序的设计

建立一个性能良好的中断环境是编好6527软件的关键。中断程序的设计一般采取如下的方法:

(1) 从0页选定3个单元 P0、P1、P2,其中 P1、P2两单元地址相连。

(2) 编写中断程序

中断程序一般采取如下的模式,见程序 No. 2-28。

No. 2-28

```
    PHA
    LDA  $P0               ;取系列中断程序入口地址偏移量
    BNE SA                 ;不等于0则转 SA 进行中断处理
    PLA
    RTS                    ;等于0则中断返回
SA  TAX
    DEX
    TXA
    ASL
    TAX
    LDA SUBDATA , X        ;取系列中断程序的地址低位
    STA  $P1
    LDA SUBDATA+1 , X      ;取系列中断程序的地址高位
    STA  $P2
```

JMP ($00P1)　　　　;转入中断处理程序

注——SUBDATA 为系列中断程序入口地址的数据地址，系列中断程序可分为读 PPU 内存、写 PPU 内存、音响发声控制等操作。至于动画处理、读操纵器状态等操作也可以安装在此处。具体程序将在以后章节的分析中看到。

另外，在中断处理程序中，一开始还必须保存 A、X、Y 寄存器的数值，返回之前恢复。

若中断处理程序中不将 P0清零，则调用一直持续，直至清零为止。显然，通过对 P0赋值，可以控制转移到不同的中断处理程序进行不同的控制操作。

4、中断状态的进入

下面通过一个实例，介绍中断状态的进入方法。

例2－7　设计一计时程序，用于检测程序的运行时间。

汇编程序见 No. 2－29(假定我们把机器语言程序安装在 $7F00开始的 RAM 中)。

No. 2－29

```
7F00  E6  D2       INC  $D2
7F02  D0  02       BNE  $7F06
7F04  E6  D3       INC  $D3
7F06  4C  0F  88   JMP  $880F
7F09  A9  00       LDA  #$00
7F0B  85  EE       STA  $EE
7F0D  A9  7F       LDA  #$7F
7F0F  85  EF       STA  $EF
7F11  60           RTS
```

说明：程序 No. 2－29适用于中断程序入口为 $880F 的版本。读者可使用

```
PRINT HEX$ ( PEEK ( &HEE ) )
PRINT HEX$ ( PEEK ( &HEF ) )
```

检查。若打印结果为0FH、88H，则可直接使用 No. 2－29；若是另外的数值，则把第一个数据置入 $7F07、第二个数据置入 $7F08，即可使用程序 No. 2－29。

调用 No. 2－29的 F BASIC 程序见 No. 2－30。

```
5 REM No. 2-30
10 FOR I = &H7F00 TO &H7F11 : READ A : POKE I , A : NEXT
20 DATA 230 , 210 , 208 , 2 , 230 , 211 , 76 , 15 , 136,169 , 0 , 133 , 238 , 169 ,
   127 , 133 , 239 , 96
30 END
65510 POKE &HD3 , 0 : POKE &HD2 , 0 : CALL &H7F09 : RETURN
65520 END
65530 A = PEEK ( &HD3 ) : B = PEEK ( &HD2 ) :
      C = A * 5 + ( A * 6 + B ) / 50 : PRINT " TIME = "C :
      RETURN
```

程序 No. 2－30的使用说明：

(1) 程序 No. 2－30共分三段：

10～30行为输入 No. 2－29程序的处理程序，该段程序运行后可删去而输入其它应用程

序；

65510行是计数器清零处理，计时开始时要先对计数器清零；

65530行为计时结果输出程序。

(2) 当需要计时时，首先调用65510子程序，对计时单元＄D2、＄D3清零，以启动计时。

(3) 当需要检查计时时间时，只要调用65530子程序即可打印出计时的时间，时间单位为秒。

关于中断程序的其它应用，我们将在以后章节中介绍。

第三章 人机对话工作系统浅析

人机对话工作系统包括：

· 复位操作过程

· 对话板工作过程

· 音乐板工作过程(MUSICBOARD)

· 计算板工作过程(CALCULATORBOARD)

· 打字板工作过程(MESSAGEBOARD)

五大部分。F BASIC 工作系统也要通过人机对话操作进入。因此，人机对话工作系统是 F BASIC BS. 2A 版本的控制中枢系统。在该系统中既有 32 列×30 行背景画面的绘制过程，又有背景画面改变色调、显示字符的工作过程；既有背景音乐的插入、控制，又有卡通图案的定义操纵。可以毫不夸张的说：人机对话工作系统较系统地体现了任天堂游戏编程的基本方法、技巧。研究分析对话工作系统的工作过程，无疑是研究任天堂游戏编程方法的最捷路径。

通过对话工作系统的分析，必然展宽我们的编程视野和拓宽编程思路。尤其是系统中的许多子程序可供我们在 BASIC 编程中直接调用，从而可较大幅度地压缩 BASIC 程序量，使有限的用户内存空间发挥较大的作用；还可以大幅度地提高 BASIC 程序的运行速度和工作效率，以较小的程序量解决较复杂的问题。因此，全面掌握、熟练运用系统子程序，可在 BASIC 编程中收到事半功倍的效果。

下面我们将分七节介绍人机对话工作系统的工作过程。

3・1 对话工作系统分析概说

在进行分析之前，若预先了解系统工作程序的特征；掌握系统 RAM 区的划分及重要工作单元的作用；弄通复位程序、中断程序的特点；并对分析讨论中的有关内容进行统一约定；必将简化讨论、帮助理解、深化认识、有利掌握。

3・1・1 系统软件的特征

人机对话工作系统软件位于 ROM 区 $C400～$FFFF 约 14K 空间。其中用于画面显示、音乐发声等数据约占 4K，程序占 8K。系统复位程序首址为 $C400，中断程序首址为 $CDA7。

3・1・2 系统工作中使用的 RAM 单元及作用

人机对话工作系统在工作中要占用大量的 RAM 单元，尤其对 0 页单元的使用更为频繁。现将使用的重要单元及作用介绍如下：

1、零页：

· $00～$0F：各类操作数据的暂存单元。

· $10：$2000 软开关状态数据暂存器。

· $11：$2001 软开关状态数据暂存器。

• $18：读键工作单元。按 Ctrl、Alt、Shift 键时置 FFH。

• $21：MUSIC 板的演奏接续器。MESSAGE 板、CAL・板的连读键标志。

• $23：羽毛笔画圈控制器。各板工作时，每输入一个字符时由该单元控制羽毛笔画一个圆圈。

• $24：MUSIC 板演奏节奏控制器。共分六档控制演奏节奏。

• $28：MES・板的卡通运动控制器。

• $2A：送显字符的显示延时器。

• $2C：复位画面中的发声控制器。

• $2D：显示延时或等待操作者输入延时器。

• $2E：操作者输入字符计数器高位。

• $2F：操作者输入字符计数器低位。

• $30：各板画面中指示卡通或卡通块的 X 坐标。在对话板中为光标，在另外三板中为羽毛笔。

• $31：各板画面中指示卡通的 Y 坐标。

• $32：操作者输入的字符送入 PPU 的地址高位。

• $33：Del、Ins 键工作单元，用于暂存 $32 单元的内容。

• $34：操作者输入的字符送入 PPU 的地址低位。

• $35：Del、Ins 键工作单元，用于暂存 $34 单元的内容。

• $36：各板工作标志。CAL・板等于 0；MES・板等于 1；对话板等于 3；MUS・板等于 4。

• $37：操作者按键键码存储器。

• $38：操作者按键键码暂存器。

• $39：操作者按键键码比较器。

• $3A：操作者输入字符有效标志。

• $3B：中断计数器。由它记录对话板工作中执行中断程序的次数。

• $3C：演奏工作单元。

• $3E、$3F：MUS・板定义三音框的工作单元。

• $40：操作者按键功能执行器。

• $41：CAL・板输入算式合法标志，也是执行计算命令标志。

• $42：MUS・板输入音符标志。上段等于 0；中段等于 1；下段等于 2。

• $43：对话板询问内容标志。

“F BASIC NEED?”时 $43=0；

“CALCULATORBOARD NEED?”询问时 $43=1；

“MUSICBOARD NEED?”询问时 $43=2；

“MESSAGEBOARD NEED?”询问时 $43=3。

• $50：小写字母键工作单元。按 Caps Lock 键时等于 FFH。

• $51～$58：随机数发生器工作单元。

• $60：执行程序入口的变址调节器。

• $61：光标形状工作单元。

• $62：$60、$6B、$C3 单元数据入口变址调节器。

· $64:$60、$6B、$C3 单元数据的地址指针。
· $66:功能键标志。按功能键时等于 FFH。
· $6B:操作性质、种类控制器。
· $6F:Ctrl 键标志。按该键时等于 FFH。
· $70:Shift 键标志。按该键时等于 FFH。
· $71:Alt 键标志。按该键时等于 FFH。
· $72:Caps Lock 标志。小写字母时等于 FFH。
· $74:读键延时器(控制稳键时间)
· $75:键码暂存器。
· $77、$78:连读键功能建立延时器。
· $79:连读键稳键延时器。
· $7B:连读键标志。
· $7C:MUS·板清行标志。
· $7D:MUS·板 Home、Home+Shift 键工作标志。
· $87:$62 单元数据转送器。
· $88:存各板名称的序号减 2 的数值,以控制显示各板的名称。
· $8A:各板画面绘制控制器。
· $8B:MUS·板演奏标志。演奏时等于 1。
· $8C:MUS·板演奏速度代号。
· $8E~$9F:CAL·板数据运算工作单元。
· $A0:演奏音符计数器,也是音符的取数变址指针。
· $A1:演奏中暂停标志(即按空格键)。
· $A3:演奏时第一音符是"$"的标志。
· $A4:反复演奏标志。读到"$"时置 1。
· $B0:CAL·板按"="键标志。
· $B1:CAL·板运算工作单元。
· $B2:MES·板检索卡通块序号的变址指针。
· $B3:MES·板存运动卡通的 X 坐标。CAL·板为小数点标志。
· $B4:MES·板运动卡通的速度控制器。CAL·板为运算符标志。向 $550 以后送数的变址指针。
· $B5:MES·板检索运动卡通块序号地址的变址指针。CAL·板为小数点标志。从 $63D 开始取数的变址指针。
· $B6:MES·板运动卡通的运动量控制器。CAL·板为送显运算结果的变址指针。复位画面中为绘图进程标志。
· $B7:对话板中为清除两栏的换行指针。
· $B8:复位画面中,各信号灯颜色控制器。
· $B9:对话板中,两栏的 PPU 地址低位。
· $BA:对话板中,两栏的 PPU 地址高位。
· $BB:对话板中,两栏下一次送入 PPU 显示字符的地址低位。
· $BC:对话板中,两栏下一次送入 PPU 显示字符的地址高位。

· $BD:CAL·板检验算式时为运算符标志。

· $BE:CAL·板检验算式时存数据的位数。

· $BF:CAL·板运算工作单元。

· $C0:对话板,COMPUTER 栏显示的各板名称的地址低位。

· $C1:对话板,COMPUTER 栏显示的各板名称的地址高位。

· $C2:对话板,COMPUTER 栏送显各板名称的程序转移标志。

· $C3:对话板,COMPUTER 栏送显字符的数据首址指针。

· $C4:对话板,COMPUTER 栏送显各板名称的数据地址变址指针。

· $C5:对话板,COMPUTER 栏送显字符的数据变址指针。

· $C6:对话板,两栏的送显、清除控制器。$C6=0 进行清除,$C6=1 进行送显。

· $C7:对话板中提示栏操作选择器。

· $C8:对话板,提示栏的送显、清除控制器。

· $D8:CAL·板运算进程标志。运算结束时等于 1;检索到 PPU 的送显地址时等于 2;运算结果送显完毕等于 3;送显结束处理完毕等于 4。

· $D9、$DA:CAL·板送显运算结果工作单元。

· $DD、$DE:CAL·板送显运算结果暂存器。

· $E0、$E1:光标形状存储器。

· $F0:第一主声部发声标志。

· $F1:第一主声部音长控制器。

· $F3:第三主声部音长控制器。(效果音)

· $F4:第二主声部更换音符变址指针。

· $F5;第三主声部更换音符变址指针。

· $F6:第二主声部音长控制器。

· $F7:第三主声部音长控制器。(旋律)

· $F8:第二主声部音量控制器。

· $F9:第三主声部旋律、音长参数暂存器。

· $FA:第三主声部发声开关。等于 0 关闭;等于 1 开启。

· $FB:$FC 数据暂存器。

· $FC:发声种类选择器和背景音乐选择器。

· $FD:第二副声部赋值工作单元。

· $FE:第三主声部效果音选择器。

· $FF:第一主声部效果音选择器。

2、一页:

一页为系统堆栈工作区。栈底地址为 $01FF,栈顶地址为 $0100。

3、二页:

二页地址为 $0200~$02FF。是定义显示卡通、动作卡通的工作区。每连续的 4 个单元定义一个卡通块。

4、三页:

· $0300~$033F:为各板送显字符的工作单元。

· $0340~$034B:为音响工作单元。

- ＄0350～＄036F：为绘制 COMPUTER 栏工作单元。
- ＄038C～＄03EF：为 CAL·板括号运算结果暂存单元。
- ＄03F0～＄03FA：为 CAL·板运算结果暂存单元。

5、四页：

- ＄0400～＄043A：为 CAL·板运算工作单元。
- ＄0480～＄04FF：为功能键功能定义区。

6、五页：

- ＄0500～＄0512：对话板存操作者输入的姓名。
- ＄0550～＄0562：对话板存操作者输入的内容。

7、六、七页：

- ＄0634～＄063C：读键工作单元。存各按键的键值。
- ＄063D～＄073D：各板工作数据暂存单元。

8、其它：

- ＄703A、＄703B：各板工作标志。对话板中＄703A＝5A、＄703B＝CC；其它三板都等于 1。
- ＄7100～＄71D7：CAL·板数据库。
- ＄7200～＄731F：MUS·板数据库。
- ＄7400～＄75FF：MES·板数据库。
- ＄7600～＄76FF：CAL·板算式存储单元。

3·1·3 读键工作过程简介

读键工作程序位于 F BASIC 系统软件 ROM 区，但在人机对话系统的工作中要经常调用它，因为操作者的字符输入和屏幕编辑等项操作的按键都要通过读键程序读出键码，以进行各种判断处理。因此，在分析对话工作系统之前，必须首先了解读键程序的工作过程和运行结果。

一、读键程序

读键程序位于 ROM 区＄AF88～＄B0EB。读键程序的数据区位于 ROM 区＄B0EC～＄B253。（程序清单见附录）

1、读键主题程序

读键主题程序在＄AF9B～＄AFE2 单元。通过这一段程序的操作，把读出的按键值存入＄634～＄63C 共 9 个单元中。（这一部分程序要反复对 CPU 的输入、输出单元＄4017、＄4016 操作，我们不必追求这段程序为什么能够读出键值，只要明确这段程序可把各按键的键值读出并存入＄634～＄63C 单元就可以了。）

各按键对应的键值列于表 3－1。

表 3－1 各按键值与所对应的存储单元一览表

按　　键	＄634	＄635	＄636	＄637	＄638	＄639	＄63A	＄63B	＄63C
Esc								40	
1								2	
2								1	
3							1		
4						1			

按　　键	$ 634	$ 635	$ 636	$ 637	$ 638	$ 639	$ 63A	$ 63B	$ 63C
5						2			
6					1				
7					2				
8				1					
9				2					
0									1
-		2							
=		8							
\	2								
Tab									80
CapsLock	8								
Shift(左)								8	
Shift(右)	4								
A							10		
B					8				
C						4			
D						10			
E							2		
F						8			
G					20				
H					10				
I				40					
J				10					
K			10						
L			20						
M				8					
N				4					
O			40						
P			2						
Q								20	
R						20			
S							20		
T						40			
U				20					
V					4				
W							40		
X							8		
Y					40				
Z							4		
Home	10								
Enter	40								
Stop	1								
Alt								4	
Ctrl								10	
Space									4

按　　键	$634	$635	$636	$637	$638	$639	$63A	$63B	$63C
Ins			1						
Del									2
;		10							
'		20							
,			4						
.			8						
/		4							
[		40							
]	20								
Ctrl+Alt								14	
↓									8
↑									10
→									20
←									40
F1								80	
F2							80		
F3						80			
F4					80				
F5				80					
F6			80						
F7		80							
F8	80								

注——1、复合键的键值为组成该复合键的所有键值之和。当各键值不在一个存储单元时则分别存放。如"@"键要按 Shift+2 键，则键值为 $63B=9(按左 Shift)或 $634=4、$63B=1(右 Shift)；再如"*"要按 Shift+8，则键值为 $637=1、$63B = 8 或 $637=1、$634=4(因按左、右 Shift 键值有区别)。

2、表中空格部分全部为零。

二、读键过程

下面分析读键过程：

1、读键程序从 $AF88 开始。假定当前没按功能键，则 $66 单元=0。由 $AF8C 调 $AF9B 子程序。

2、假定按"A"键，则由表 3－1 查出 $63A=10，其它各键值单元都等于 0(执行 $AF9B～$AFE3 后)。

3、对 $6F、$70、$71 单元置 0 后，跳过 $AFEC～$B007。

4、由 $B008 调 $B02C 键码翻译子程序。

5、由 $B02C～$B038 程序检索，当 X=6 时，A=1，故转 $B060。

6、把 X=6 送入 $6E 单元后，由 $B062～$B06B 程序段进行移位操作，当 Y=4 时 C=1，转 $B06C。

7、把 Y=4 送入 $6D，令 X 乘 8(连续左移三次)再加 $6D 单元内容，因而 X=34H。

8、因为未按 Shift、Ctrl+Alt、CapsLock 键，所以 $6F= $70= $71= $72=0，因此 A=($B0EC+X)=($B0EC+34)=($B120)=41 返回 $B008。

9、因 A=41 转 $B010。因未按其它操作键故 $18=0、$50=0，因此返回 $AF8C。

10、因 A=41<F0H 则返回调用程序。

至此，读键过程结束。字母“A”的 ASCII 码“41”存于 A 寄存器中。

至于其它按键 ASCII 码的翻译过程，读者只要按照表 3－1 列出的键值顺藤摸瓜，都能极容易地翻译出按键的 ASCII 码。故在以后的讨论中，当调用读键程序时，我们将不再赘述读键过程而直接给出按键的 ASCII 码。

3·1·4 系统分析中的统一约定

为使以后的讨论规范化，特对程序分析中的某些术语和表示方法约定如下：

1、凡分析讨论中的数据，一般均采用 16 进制。对单独的 16 进制数据均冠以前缀“H”，如 16 进制数据“20”表示为 H20，以与 F BASIC 编程书写方式相统一；对于算式或等式中的 16 进制数据则不加任何标志，以简化书写；对于 10 进制数据或小于 H0A 的 16 进制数据则不加任何标志以符合记数习惯（注——算式、等式中不用 10 进制数）。

2、程序或数据所在内存单元地址的表示，采用汇编语言中的表示方法。即 16 进制数据均冠以前缀“＄”符号。一般用四位数表示，如＄3F00、＄EFB3 等。当高位为 0 时则省略以简化书写，如＄0007 写为＄07 或＄7、＄063D 写为＄63D 等。

3、对背景图形库中的图形代号，不再使用原来的称呼（在拙作《F BASIC 语言与编程技巧》中的代号称呼）而直接称其在图形库中的序号，以与其对应的 ASCII 码兼容，便于在机器语言编程中直接使用。如原“A1”号图形则称为 0 号图形、原“M8”号图形则称为 HFF 号图形。

4、为便于讨论，各板名称采用如下简写形式：

人机对话板———MC·板

音乐板(MUSICBOARD)———MUS·板

计算板(CALCULATORBOARD)———CAL·板

打字板(MESSAGEBOARD)———MES·板

5、在读键过程中，我们把经＄AF9B～＄AFE3 处理后得出的存在＄634～＄63C 单元中的数值称为键值；把通过＄B008～＄B02C 处理后得出的按键 ASCII 码则称为键码。因此，键值和键码具有不同的含义。

6、我们知道，在 F BASIC 状态下，每一个卡通可定义为 8×8 点阵或 16×16 点阵。在以后的讨论中，我们把 8×8 点阵的卡通称为一个卡通块或叫做 SP 块（实际上，8×8 点阵的 SP 大多数是一个卡通的四分之一部分，并非完整的卡通，故称其为 SP 块较贴切。）。因此，二页中每连续四个单元定义一个 SP 块。同时，在对话工作系统中，SP 块使用得较多，称 SP 块将方便讨论。注意“卡通”和“SP”两者是等效的。

7、系统名词

我们把对话工作系统中各板的名称、各板名称的简称、以及“OK、NO、HELLO、END”提示等英文词汇统称为系统名词。

3·2 复位画面的绘制

复位画面（见封三）由上至下共分四层，其结构特点如下：

·第一层：有 24 个由 H1C 号图形构成的信号灯，共分三组，每组八个，分上、下两层排列。工作中，这 24 个灯不停地闪烁和改变色彩。

·第二层：是电脑工作栏。在该栏的上框中有阴体字“COMPUTER”标题；栏中有20列×8行共160个方块的空间，开机后为不停变化位置的三基色方块图案；对话时，显示电脑询问和表述的内容；该栏的下框右端有四个白色方块，工作中间歇性的闪动。

·第三层：是操作者工作栏。在该栏的上框中有阴体字“OPERATOR”标题；栏中有20列×3行空间，开机时为全蓝色；下框右端也有四个小方块闪动。

·第四层：是提示栏。共横向分四段，每段有5个空间，开机时四段都置为蓝色；在每一段的上方分别有“F1、F2、F3、F4”标题，标志着F1～F4功能键所代表的含义。

·在画面的左、右下角各有5个信号灯（也由H1C号背景图形构成）。工作时，这10个信号灯也随上面的24个灯一起无规律地闪烁。

·伴随着开机画面中图块、信号灯的闪烁，还发出间歇性的音响。音响特点为：开始是连续的“嘟嘟”声；随后是无规则的“滴滴哒哒”声；一段时间后又变为另一音调的“嘟嘟”声；最后无声。间歇一段时间后又重复发声。这一发声过程一直持续到按任一键后才停止。

3·2·2 复位状态的建立

我们知道，复位程序的入口地址由\$FFFC、\$FFFD指出。查这两个单元的内容，可得出复位程序的入口地址为\$C400。执行\$C400程序，共进行以下几方面的操作：

(1) 令\$2000=10，关中断，封锁键盘。背景选用图形库第Ⅰ区图形。

(2) 设置堆栈顶。

(3) 读中断。

(4) 由\$C411～\$C422程序段处理，令\$0000～\$07FF清零，进行系统RAM初始化。

(5) ·令\$7A=HFF，由\$C427调\$CD94，令\$4011=0，关第二副声部。

·令\$2001=6，关背景、卡通页显示。

·把\$0200～02FF置为HF8，清卡通显示页。

·令\$12=0、\$13=0、\$10=0。

·令2001=0，关显示。

·置PPU地址为\$2000，令\$2000～\$23BF=20，\$23C0～\$23FF=0，清背景0页。

·返回\$C427。

(6) 令\$E0=HFD、\$E1=H5F，设定光标的两种形状。

(7) 令A=H10 EOR H80=H90送入\$2000，开中断，选通键盘。

(8) 令\$4015=0F，开三个声部和第一副声部发声开关。

(9) 令\$ED=H4C、\$EE=HA7、\$EF=HCD，置中断程序入口为\$CDA7。

至此，复位工作状态建立完毕。

3·2·3 对话系统主控程序简介

对话工作系统的主控程序位于系统ROM区\$C44A～\$C477。为便于分析，特把主控程序及各程序段的作用注释列于程序No.3－1。

No.3－1 对话系统主控程序

```
C44A  A560     LDA  $60    ┐
C44C  C901     CMP #$01    │ $60≠1 时转 $C45A
C44E  D00A     BNE  $C45A  ┘
C450  A510     LDA  $10    ┐
C452  297F     AND #$7F    │ $60=1 进行进入 F BASIC 系统处理。
C454  207AEC   JSR  $EC7A  │
C457  4C0080   JMP  $8000  ┘
C45A  20E8ED   JSR  $EDE8  ┐
C45D  A541     LDA  $41    │ $41=0 转 $C475(即没有发出运算命令则转移)
C45F  F014     BEQ  $C475  ┘
C461  20A6DF   JSR  $DFA6  ┐
C464  A5B9     LDA  $B9    │
C466  F006     BEQ  $C46E  │
C468  2001DE   JSR  $DE01  │ $41≠0 进行 CAL·板的运算处理。
C46B  4C71C4   JMP  $C471  │
C46E  20A9DD   JSR  $DDA9  │
C471  A900     LDA #$00    │
C473  8541     STA  $41    ┘
C475  4C4AC4   JMP  $C44A
```

显而易见,对话系统中,除 CAL·板外,若 $60≠1 则主控程序将有程序 No. 3－2 的形式:

No. 3－2

```
C441 A560      LDA  $60
C45A 20E8ED    JSR  $EDE8
C475 4C4AC4    JMP  $C44A
```

这时,对话主控程序的工作过程仅仅是反复地调用 $EDE8 随机数发生子程序,令 $51～58 单元产生随机数。因此,在以后的讨论中除 CAL·板外,我们仅对 No. 3－2 的程序进行分析。

3·2·4 对话系统的中断处理

对话工作系统的中断处理程序位于 ROM 区 $CDA7～$CE45。

按功能可把中断处理程序划分为五个部分:

1、现场保护程序($CDA7～$CDB3)

$CDA7～$CDB3 是中断现场保护处理程序,其作用是在进行中断处理前,把 P、A、X、Y 四个寄存器的当前值入栈保护;把 $00～$0F 单元的内容入栈保护。每执行一次中断,都要处理一次。

2、预处理程序($CDB4～$CDD2)

$CDB4～$CDD2 是中断预处理程序,其作用是进行正式中断处理前的有关预备工作处理。它共进行以下几项操作:

· 令 $10 AND H7F=H10 送入 $2000 和 $10 单元,关中断。

· 令 $ 11 OR H1E=H1E 送入 $ 2001 和 $ 11 单元，打开背景页、卡通页显示开关。

· 令 $ 2003=0，设置 DMA 传送首地址低位。

· 令 $ 4014=2，设置 DMA 传送首地址高位。

以上操作，每次中断都要处理一次。

3、中断主控程序（$ CDD3～ $ CE28）

$ CDD3～ $ CE28 是对话工作系统的中断处理主程序，各板的画面绘制和操作几乎都由该段程序控制完成。因此，它将是我们今后讨论的重点。

4、结束处理程序（$ CE29～ $ CE36）

$ CE29～ $ CE36 是中断结束处理程序。它主要进行以下三项操作：

· 读 $ 2002 软开关。

· 令 $ 10 EOR H80=H90 送入 $ 2000、$ 10，开中断。

· 令 $ 7A=0 以封闭送显示操作。

以上三项操作，每中断一次都处理一次。

5、现场恢复程序（$ CE37～ $ CE45）

$ CE37～ $ CE45 是中断现场恢复处理程序。在中断处理结束、返回主程序之前，要恢复执行中断处理前的现场，以使主程序继续运行。主要操作是：从堆栈取回 $ 00～ $ 0F 和 P、A、X、Y 寄存器的值，使其恢复到中断前的状态。

至此，一次处理中断结束。

我们已经知道：6527CPU 的中断周期为 0·02 秒，也就是说，每秒钟 CPU 要对以上中断处理的五项操作进行 50 次。可见，中断处理是极其频繁的。就因为如此，对话工作系统才把大部分工作过程安置在中断处理中进行。因此，中断处理将是我们讨论的重点之重。

3·2·5　中断处理主控程序分析

众所周知，人们每天早晨起床、晚上睡觉已是正常的生活习惯，毫不为奇。考察某个人的政绩、评价某人的工作能力，也没有必要调查他的起床、睡觉过程。原因是：起床、睡觉虽然是人生不可缺少的活动，但它与人们的工作关联并非密切。因此，在考察政绩、能力时可略去不计。

我们分析研究中断处理过程也是这样。通过以上讨论我们已经清楚：中断处理中的第 1、2、4、5 步就如同人们的起床、睡觉一样，每次中断都要例行处理一次，因此我们可以从中断处理程序中单独抽出第 3 部分进行分析研究，而对于剩余的部分只要别忘记它们是中断处理的组成部分和它们所完成的操作就可以了。由此形成了中断处理的主控程序。

另外，为使中断主控程序清晰易读，还可以删除其中的次要部分，仅在与主控程序的工作有关联时再进行分析。这样，我们得到了简化的中断处理程序 No. 3－3。

No. 3－3 中断主控程序

```
CDD3   A68A     LDX $8A          ┐
CDD5   BD69CA   LDA $CA69,X      │
CDD8   8500     STA $00          │
CDDA   BD73CA   LDA $CA73,X      │
CDDD   8501     STA $01          │
CDDF   2043EE   JSR $EE43        │ 对背景页操作,完成各板画面的绘制和字符的送显。
CDE2   A900     LDA #$00         │
CDE4   858A     STA $8A          │
CDE6   8D0003   STA $0300        │
CDE9   8D0103   STA $0301        ┘
CDFF   E63B     INC $3B            ;中断计数器加一
CE09   200380   JSR $8003        ┐
                (JMP $AF88)      │
CE0C   8537     STA $37          │ 读键
CE1C   C903     CMP #$03         │
CE1E   F009     BEQ $CE45        ┘
CE20   20D2ED   JSR $EDD2          ;各计数器减一
CE23   2006FC   JSR $FC06          ;发声处理
CE26   206ECE   JSR $CE6E          ;转移程序入口
CE45   40       RTI                ;中断返回
```

说明:

一、在中断主控程序中删去的几段程序功能简介:

1、$CDEC～$CDF2

这段程序的作用是:在询问对话中,当选择“F BASIC”时,控制转入“绘制 F BASIC 选项画面的操作程序”。

2、$CDF3～$CEFE

这段程序的作用是:在询问对话中,等待操作者输入的延时计数器为 0 时,返回复位画面的转移处理。

3、$CE01～$CE08

这段程序的作用是:当随机数启动器 $51 单元等于 0 时,令 $51=H0F,以启动随机数发生器开始工作。

4、$CE0E～$CE1B

这段程序的作用是:根据 $7A 的取值决定是否转移。若 $7A=0 则跳过该段程序。因在中断处理中,$7A 常为 0,所以可以删去。

二、中断主控程序分析

我们仅讨论 No. 3－3。

中断主控程序 No. 3－3 可按功能划分为三大段:

1、背景页操作程序 $CDD3～$CDEB

这段程序根据 $8A 的取值,可对背景页面进行不同的操作。它的主要任务是完成各板字符的送显和各板背景画面的绘制。

2、读键程序＄CE09～＄CE1F

这段程序的作用是:读出键码;截获 STOP 按键。当按 STOP 键时则控制中断返回。

3、各板工作程序＄CE20～＄CE28

这段程序完成各板的工作过程控制和各种操作。它包含三个子程序。作用分别如下:

＄EDD2 子程序。作用是控制各计数器减一

＄FC06 子程序。作用是控制各声部发出效果音响或音乐旋律。

＄CE6E 子程序。作用是设置各种处理程序的转移入口,并控制转入各处理程序。

3・2・6 重要子程序介绍

为减少以后分析讨论的工作量,我们首先预习经常调用的几个子程序。

1、随机数产生子程序＄EDE8

从 No.3－2 可见,＄EDE8 是对话系统主控程序的主要工作程序。它的作用是通过对＄51～＄58 单元进行有条件的循环移位,产生出一系列随机数。

随机数的产生过程如下:

开机后,由中断程序对＄51 单元赋以初值,有＄51＝H0F。以后通过子程序的自身运行,使＄51～＄58 八个单元中存有不同的随机数。

随机数的用途是:

使复位画面"COMPUTER"栏中的三基色方块不停地、无规律地变换位置。

使复位画面中的"滴滴哒哒"声响无规律地发出,永不重复。

使对话板中的 34 个信号灯随机闪烁,变幻莫测。

控制 CAL・板、MES・板工作中的背景音乐旋律各乐章随机组合,从而以较少的音乐数据演奏出较复杂的音乐旋律。

＄EDE8 子程序教给我们一种获取随机数的技巧。它比 F BASIC 中通过 RND 函数获取随机数的方法更简单、更奇妙。它是在机器语言编程中获得随机数的有效方法。特提请读者重视。

为便于以后各节的讨论和对随机数的使用,特列出开机后自＄51＝H0F 开始,执＄EDE8 子程序第 100～119 次、200～219 次、300～319 次后各单元的数据于表 3－2。

表 3－2 执行＄EDE8 子程序所获得的随机数表

执行次数	＄51	＄52	＄53	＄54	＄55	＄56	＄57	＄58
100	B5	B4	0E	C6	DB	56	E0	4D
101	5A	B2	07	63	6D	AB	70	26
102	2D	59	03	B1	B6	D5	B8	13
103	16	AC	81	D8	DB	6A	DC	09
104	8B	56	40	EC	6D	B5	6E	04
105	45	AB	20	76	36	DA	B7	02
106	A2	D5	90	3B	1B	6D	5B	81
107	D1	6A	C8	1D	8D	B6	AD	C0
108	E8	B5	64	0E	C6	DB	56	E0
109	74	5A	B2	07	63	6D	AB	70
110	BA	2D	59	03	B1	B6	D5	B8
111	DD	16	AC	81	D8	DB	6A	DC

执行次数	$51	$52	$53	$54	$55	$56	$57	$58
112	EE	8B	56	40	EC	6D	B5	6E
113	77	45	AB	20	76	36	DA	B7
114	BB	A2	D5	90	3B	1B	6D	5B
115	5D	D1	6A	C8	1D	8D	B6	AD
116	2E	E8	B5	64	0E	C6	DB	56
117	97	74	5A	B2	07	63	6D	AB
118	CB	BA	2D	59	03	B1	B6	D5
119	65	DD	16	AC	81	D8	DB	6A
200	98	87	B6	B9	D4	A7	0E	40
201	CC	43	DB	5C	EA	53	87	20
202	E6	21	ED	AE	75	29	C3	90
203	F3	10	F6	D7	3A	94	E1	C8
204	F9	88	7B	6B	9D	4A	70	E4
205	7C	C4	3D	B5	CE	A5	38	72
206	3E	62	1E	DA	E7	52	9C	39
207	1F	31	0F	6D	73	A9	4E	1C
208	8F	98	87	B6	B9	D4	A7	0E
209	C7	CC	43	DB	5C	EA	53	87
210	E3	E6	21	ED	AE	75	29	C3
211	71	F3	10	F6	D7	3A	94	E1
212	B8	F9	88	7B	6B	9D	4A	70
213	5C	7C	C4	3D	B5	CE	A5	38
214	2E	3E	62	1E	DA	E7	52	9E
215	17	1F	31	0F	6D	73	A9	4E
216	0B	8F	98	87	B6	B9	D4	A7
217	05	C7	CC	43	DB	5C	EA	53
218	82	E3	E6	21	ED	AE	75	29
219	41	71	F3	10	F6	D7	3A	94
300	D6	AB	06	50	5C	FC	45	BD
301	6B	55	83	28	2E	7E	22	DE
302	B5	AA	C1	94	17	3F	11	6F
303	DA	D5	60	CA	0B	9F	88	B7
304	ED	6A	B0	65	05	CF	C4	5B
305	F6	B5	58	32	82	E7	E2	2D
306	FB	5A	AC	19	41	73	F1	16
307	7D	AD	56	0C	A0	B9	F8	8B
308	3E	D6	AB	06	50	5C	FC	45
309	1F	6B	55	83	28	2E	7E	22
310	0F	B5	AA	C1	94	17	3F	11
311	87	DA	D5	60	CA	0B	9F	88
312	43	ED	6A	B0	65	05	CF	C4
313	A1	F6	B5	58	32	82	E7	E2
314	D0	FB	5A	AC	19	41	73	F1
315	E8	7D	AD	56	0C	A0	B9	F8
316	74	3E	D6	AB	06	50	5C	FC

执行次数	$ 51	$ 52	$ 53	$ 54	$ 55	$ 56	$ 57	$ 58
317	BA	1F	6B	55	83	28	2E	7E
318	5D	0F	B5	AA	C1	94	17	3F
319	AE	87	DA	D5	60	CA	0B	9F

由于主控程序每秒钟就要运行近万次，所以采用这种方法获得的随机数具有较可靠的随机性。表 3－2 就有力地证明了这一点。

2、计数器减一子程序 $ EDD2

$ EDD2 子程序的作用是：每中断一次令 $ 21～$ 2A 单元的内容减一；每当中断 20 次令 $ 2B～$ 2D 单元的内容减一。当某一单元为 0 时将不再减少，维持 0 值。

调用该子程序的目的是：令中断处理中的计数器减一，达到对显示延时、羽毛笔画圈、音符发声的音长、音响变换间隔、等待操作者输入延时等方面的控制。

以上两个子程序的工作过程较简单，也很容易理解。因此在今后的讨论中，当调用它们时将直接使用运行结果而不再讨论运行过程。

3、程序入口转移控制子程序 $ CE6E

$ CE6E～$ CED4 是各类操作程序入口的转移控制子程序。人机对话系统中的各类操作都要通过该子程序控制转移到相应的程序入口。因此，该子程序在人机对话系统工作中占有举足轻重的地位。

该子程序共分四段，各段的作用如下：

(1) 主控程序 $ CE6E～$ CE81

通过调用 $ CE82 子程序，从数据区 $ CB88～$ CD03 中查出对应的数据赋给 $ 60 单元；由 $ 60 变址，从程序入口数据区 $ CD43～$ CD92 中查出程序的入口地址，转入相应的处理程序。

(2) 对 $ 60 赋值控制子程序 $ CE82～$ CE8A

$ CE82 是向 $ 60 单元赋值的控制子程序。它首先检查 $ 60 单元的当前值，若 $ 60≠0 时则令 $ 73＝0，转原程序入口；当 $ 60＝0 时，则进行重新赋值处理。

(3) $ 60 赋值程序段 $ CE8B～$ CEC8

$ CE8B～$ CEC8 是为 $ 60、$ 6B、$ C3 三个单元赋值的专用程序。赋值操作以 $ 62 单元为数据区首址指针，从 $ CB28～$ CB8F 单元查出数据区首址；然后以 $ 64 单元为变址指针，从数据区 $ CB88～$ CD03 中查出数据分别赋给 $ 60、$ 6B、$ C3 单元，供计算程序转移入口和决定操作种类使用。由 $ 62 单元取值决定的 $ 60 数据区首址对应情况列于表 3－3。

表 3—3　$ 62 单元取值与 $ 60 单元数据区首址对照表

$ 62	$ 60 数据区首址	$ 62	$ 60 数据区首址	$ 62	$ 60 数据区首址
00	CB88	10	CC2E	20	CCAF
01	CB8E	11	CC3A	21	CCB4
02	CB92	12	CC3A	22	CCB9
03	CBBA	13	CC3F	23	CCB9
04	CC15	14	CC3F	24	CCBE
05	CBC6	15	CC48	25	CCBE
06	CBCB	16	CC48	26	CCBE
07	CBCB	17	CC48	27	CCBE

$62	$60 数据区首址	$62	$60 数据区首址	$62	$60 数据区首址
08	CBE6	18	CC4B	28	CCDA
09	CBE6	19	CC92	29	CCEC
0A	CBEB	1A	CC92	2A	CCE6
0B	CC00	1B	CBD8	2B	CCDD
0C	CC07	1C	CC9B	2C	CCBE
0D	CC0B	1D	CC9F	2D	CCFD
0E	CC22	1E	CBE4	2E	CCFD
0F	CB98	1F	CCA7	2F	CBA1

(4) $62 单元数值调整程序 $CEC9～$CED2

$CEC9～$CED2 为调整 $62 单元数值的程序段，主要作用是调整 $62 单元的赋值，以控制 $60 单元选择数据区首址。

为便于讨论，特列出 $60 单元取值对应的程序入口和该程序的作用对照情况于表 3－4

表 3－4　$60 单元取值与程序区入口和作用对照表

$60	程序入口	主　要　作　用
01	CD93	直接返回
02	DC77	CAL・板工作程序
03		空
04	CF5B	根据 $703A、$703B，对 $88、$64 单元赋值，令 $60＝0
05	CED5	令 $2D＝$6B，置等待操作者输入延时
06	D2A4	对两栏操作
07	CF78	等待操作者输入
08	CEF8	关显示。令 $78＝1、$8A＝9、$60＝0，绘 MUS. 画面
09	CF06	清屏。清二页。令 $60＝0
0A	D515	复位画面的绘制与等待操作者按键处理
0B	DB97	MES. 板工作程序
0C	D44A	对提示栏操作
0D	CF1F	把 $63D～$64F 的内容送入 $500～$512。令 $60＝0
0E	CEF8	令 $78＝1、$8A＝$6B、$60＝0
0F	CF18	令 $78＝0、$60＝0
10	D0E8	翻译功能键等
11	D02A	对 $7000～$7FFF 有关单元操作
12	D107	令 $62＝$87
13	CFC4	对话中各系统名词的翻译、检索、标志存入 $00 单元中
14	CF2D	对 $7000～$7FFF 有关单元操作
15	D06E	读入操作者输入的内容
16	CF10	清二页，令 $60＝0
17	D0DA	令 $43＝0、$88＝0F、$60＝0、$80～$86＝0
18	F634	$D0～$D9→$A0～$A9、$530～$539→$520～$529
19	FA0D	翻译功能键
1A	F646	检查功能键
1B	F67A	令 $FE 等于 H10 或 H20
1C	D10F	定义 CAL. 板功能键、羽毛笔
1D	D124	MES. 板定义羽毛笔

$60	程序入口	主　要　作　用
1E	F4B1	定义功能键,$60=0
1F	CEE8	令$82=12、$60=0,调整$64
20	D10C	令$62=1E、$87=7、$43+1、$88=$43+0F、$64=$60=0
21	F689	CAL.板工作子程序
22	F98E	令$81=0、$2D=0、$82=18、$AE=FE、$60=0、$64=0
23	F9AF	令$FC=3、对$60、$C3赋值
24	F9E2	令$60=0、$00=0、调整$64、检查$DA~$DC
25	D139	定义MUS.板功能键、SP
26	E423	MUS.板工作程序
27	D103	令$87=$45、$62=$45、$64=0、$60=0

3·2·7 复位画面中背景页图形的绘制

复位画面使用了背景、卡通两个显示页面。其中两栏下框右端的八个白色方块、阴体字"COMPUTER、OPERATOR"、黑体字"1、2、3、4"及信号灯闪烁图案均显示在卡通页,其它部分则在背景页显示。本节主要讨论背景页图形的绘制过程。

通过以上几节的介绍,我们已了解到:对话系统的工作过程集中在中断处理中,主程序的作用仅仅是产生随机数。因此,我们仅讨论中断处理过程。(当使用随机数时,可从表3-2中任取一组使用,而不再从主程序调用随机数子程序进行推算。这是因为主程序在每秒钟要循环上万次,所以任取一组都具有代表性。)

一、清背景0页、卡通页

由于复位画面由背景、卡通两个页面叠加而成,所以画面绘制前要先清背景0页PPU映射单元和卡通图形映射单元。即清PPU的$2000~$23FF和CPU的$200~$2FF单元。

清除操作在开机后第一次中断处理中完成。步骤如下:

1、由$CE26调程序入口转移子程序$CE6E;由$62=0查表3-2得$60单元的数据首址为$CB88;对$60单元赋值,令$60=9;查表3-4设置程序转移入口为$CF06。

2、转$CF06清屏子程序。令PPU的$2000~$23BF=H20、$23C0~$23FF=0,令CPU的$200~2FF=HF8,令$60=0。

二、绘图准备

在第二次中断,查表3-3、表3-4,令$60=8,设置程序转移入口为$CEF8。

经$CEF8子程序处理,令$2001=0,关闭各显示页面;令$8A=$6B=1,为下一次中断绘图作好准备。

三、背景页绘图

背景页绘图包括:对PPU配色单元$3F00~$3F1F置入颜色代码;对背景0页配色单元$23C0~$23FF置入配色代码;对背景0页的图形单元$2000~$23BF置入图形数据。

背景页绘图在第三次中断完成。因为$8A=1,所以在执行$CDD3中断程序时为$00、$01两单元赋值,令$00=HF4、$01=HD7,设置了图形数据区首址为$D7F4。然后调用$EE43绘图子程序绘图。

绘图子程序位于ROM区$EE09~$EE56单元。它是背景页绘图的一个重要程序,也是游戏编程特技中示范程序之一。为此,特复录绘图子程序于No.3-4。

No.3-4 绘图子程序

```
EE09  8D0620  STA $2006        ┐
EE0C  C8      INY              │
EE0D  B100    LDA ($00),Y      │ 设置 PPU 首址
EE0F  8D0620  STA $2006        ┘
EE12  C8      INY              ┐
EE13  B100    LDA ($00),Y      │
EE15  0A      ASL              │
EE16  48      PHA              │
EE17  A510    LDA $10          │
EE19  0904    ORA #$04         │
EE1B  B002    BCS $EE1F        │ 确定送数方式
EE1D  29FB    AND #$FB         │
EE1F  207AEC  JSR $EC7A        │
EE22  68      PLA              │
EE23  0A      ASL              │
EE24  9003    BCC $EE29        │
EE26  0902    ORA #$02         │
EE28  C8      INY              ┘
EE29  4A      LSR              ┐
EE2A  4A      LSR              │
EE2B  AA      TAX              │
EE2C  B001    BCS $EE2F        │
EE2E  C8      INY              │ 向 PPU 送数
EE2F  B100    LDA ($00),Y      │
EE31  8D0720  STA $2007        │
EE34  CA      DEX              │
EE35  D0F5    BNE $EE2C        ┘
EE37  38      SEC              ┐
EE38  98      TYA              │
EE39  6500    ADC $00          │
EE3B  8500    STA $00          │ 数据区地址指针换算处理
EE3D  A900    LDA #$00         │
EE3F  6501    ADC $01          │
EE41  8501    STA $01          ┘
```

```
EE43   AE0220   LDX $2002        ┐
EE46   A000     LDY #$00         │
EE48   B100     LDA ($00),Y      │
EE4A   D0BD     BNE $EE09        │
EE4C   A512     LDA $12          │ 向PPU送数的开始和结束处理
EE4E   8D0520   STA $2005        │
EE51   A513     LDA $13          │
EE53   8D0520   STA $2005        │
EE56   60       RTS              ┘
```

程序No.3－4是一个结构最优化的查表法绘图子程序。虽然程序很短小，但却包含了6527CPU与PPU通信的全部操作。按功能可把它划分为两大部分：

$EE09～$EE41：向PPU送数处理程序。

$EE43～$EE56：对PPU进行开关操作处理程序。

下面结合背景页图形的绘制过程对程序No.3－4分析如下：

1、执行$EE43程序，打开PPU操作开关。

2、根据$00、$01单元指示的数据区首址$D7F4取出第一个数据存入A寄存器，有A＝H3F。因此A≠0，故转送数处理程序$EE09。

3、送数处理

送数处理程序由四部分组成。它们的工作过程如下：

(1) 设置PPU送数首址处理程序$EE09～$EE11

这段程序的作用是从数据区中取出第一、第二两个数据送入$2006，设置PPU的通信首址。

(2) 送数方式判断处理程序$EE12～$EE28

这段程序决定向PPU的送数方式。它不仅能够决定一次向PPU送a个不同的数还是送a个相同的数，还能决定向PPU送数的取向，即是横向送数还是纵向送数。能够灵活调整向显示页面送数的走向，是6527CPU的又一大特点。后面将专题介绍送数方式的特点和工作过程。

(3) 送数程序$EE29～$EE35

这段程序的作用是从数据区取数送入PPU，并自动控制送数数量，达到送数数量后自动转出送数程序。

(4) 数据区地址指针换算程序$EE37～$EE41

当一次送数完毕后，要及时调整$00、$01单元中的数据区指针，使其指向下一次送数的数据区首址。$EE37～$EE41是专为换算数据区首址而设置的程序。

4、送数方式讨论

送数处理程序$EE09全面体现了6527CPU的送数特点。它也是任天堂游戏中背景画面的重要绘制手段。现予以详细分析。

在绘图数据区$D7F4～$DB2D中，开始的三个数据就是核心数据，我们给它起一个名字叫送数指南。送数指南中的前两个数据决定向PPU置数的首址；第三个数据决定送数方式。现在重点讨论第三个数据。(注——在PPU数据区中，凡底下划横线的三个数据即为送数指南。为方便叙述，我们把这三个数据依次编号为S1、S2、S3。)

(1) S3各位的作用

S3 各位的作用见图 3－1

D7	D6	D5	D4	D3	D2	D1	D0
送数方式		送数数量 a					

图 3－1　S3 各位含义示意图

图 3－1 中 D0～D5 位指示送数的数量 a，a 的最大值为 H3F，即一次最多可以送 H3F 个数据；D6、D7 位决定送数方式。因送数方式由两位二进制数表示，所以共有四种送数方式。送数方式代码和送数方式的对应关系见表 3－5。

表 3－5　PPU 送数方式和方式代码的对应表

方式代码		送　数　方　式
D7	D6	
0	0	横向送 S3 后 a 个单元的数据，每送一数 PPU 地址加 1 。
0	1	横向送 S3 后第一个单元的数据 a 次，每送一数 PPU 加 1。
1	0	纵向送 S3 后 a 个单元的数据，每送一数 PPU 地址加 H20。
1	1	纵向送 S3 后第一个单元的数据 a 次，每送一数 PPU 加 H20

下面结合程序的执行，讨论各种送数方式的实现。

・第一组送数指南地址在＄D7F4，数据为：

3F　00　20

——　——　——

S1　S2　S3

S1、S2 表示送数的 PPU 首地址为＄3F00，即 PPU 的配色区首址。S3＝H20，其 D7、D6 位为 00，由表 3－5 可知道这意味着把 S3 后的 H20 个单元的数据送入＄3F00～＄3F1F。以后，每送一数 PPU 地址自动加一，因此在执行＄EE12 以后的程序时，有

（＄10 OR 4）AND HFB＝H10→＄2000

由于＄2000 的 D2 位等于 0，所以为横向送数方式。

因 A＝H20，左移两次后，A＝H80；仍有 C＝0，所以转＄EE29，令 A 右移两次，恢复 S3 原值，送入计数器 X；由 Y＝Y＋1 控制，把＄D7F7～＄D816 的颜色代码依次送入 PPU 的＄3F00～＄3F1F 中。

当 X＝0 时，本次送数结束。通过数据区地址指针换算程序＄EE37～＄EE41 的处理，令＄00＝H17、＄01＝HD8，即数据区地址指向＄D817，进入下一轮取数判断、送数处理。

・第二组送数指南地址为＄D817，数据为：

23　C1　46

——　——　——

S1　S2　S3

置 PPU 首址为＄23C1。

现在 S3 的 D6＝1，由表 3－5 可知这意味着对下一单元＄D81A 中的数据要送 6 次。我们看一下在程序中是如何控制的。

执行＄EE12 以后的程序，有

A＝＄D819＝H46

A 左移一位后，A＝H8C、C＝0，故仍有＄2000＝H10，置横向送数方式。

A 左移一位后有 A＝H18、C＝1，则有

A＝H18 OR 2＝1A

Y＝Y＋1＝3

令 A 右移两次，返回原值。有 A＝6、C＝1，因而转＄EE2F，跳过了令 Y 加一的处理程序段，直接从＄D817＋Y＝＄D81A 中取数 HAA，送入 PPU。令 X－1 后，转＄EE2C，仍由于 C＝1，跳过 Y 加一的处理，仍取＄D81A 的数据 HAA 送入 PPU。从而达到了令 S3 后第一个单元的数据连送 6 次的目的。向＄23C1～＄23C6 中都置入了 HAA。

以后七组的送数情况与前两次相同（分析从略）。

· 第十组送数指南的地址为＄D850，数据为：

20	00	DE
S1	S2	S3

显然，从本组开始，对背景 0 页的图形单元置数。因 S3＝HDE，即 S3 的 D7、D6 位等于“11”。由表 3－5 可知这意味着向 PPU 纵向送 S3 后的第一个数据 HFE，连送 H1E 次，即占满背景 0 页的第一列。

下面我们分析程序的送数控制过程。

执行＄EE12、Y＝2、A＝＄D850＋Y＝HDE

A 右移一位后 A＝BC、C＝1

A 值入栈

令 A＝＄10 OR 4＝H14

现在，因 C＝1，故跳过＄EE1D，令 A 与 HFB 进行“与”运算后得 H14 送入＄2000。我们知道＄2000 开关组的 D2 位是纵向送数开关，现在该开关置一（被打开），表示为纵向送数方式。

从堆栈弹出 A 值，A＝HBC

A 左移一位有 A＝H78、C＝1

因 C＝1 所以 A＝H78 OR 2＝H7A

令 Y＝Y＋1＝3

令 A 右移两位 A＝H1E→X，C＝1

由于 C＝1 故转＄EE2F，跳过 Y＋1 的操作。

取＄D853 的数据 HFE 送入 PPU。然后，X 减一，CPU 自动控制 PPU 的地址加 H20，使下一 PPU 的送数地址为＄2020。转回＄EE2C。送下一个数……。直到 X＝0，本次送数结束。从而令背景 0 页的第一列 H1E 个单元全部置入 HFE。

· 当送到第三十五组数据时，送数指南地址为＄D8B6，送数指南为：

20	C3	88
S1	S2	S3

现在 S3 仅有 D7＝1，由表 3－5 可知这意味着从 PPU 的＄20C3 开始，连续纵向置入 S3 后的 8 个数据。

下面分析程序的控制过程：

执行＄EE12、Y＝2、A＝＄D8B6＋Y＝H88

A 左移一位后 A＝H10、C＝1

因 C=1、所以 $ 2000=H14,置纵向送数方式。

A 左移一位,A=H20、C=0

因 C=0、转 $ EE29、令 A 右移两位、A=8、送入 X、C=0

Y=Y+1=3、A= $ D8B6+Y=3→ $ 2007

X=X-1=7、转 $ EE2C

Y=Y+1=4、A= $ D8B6+Y=4→ $ 2007

……

直至 X=0,本次送数结束。从而把 S3 后的八个数据纵向送入了背景 0 页的八个单元。

以上分别讨论了程序 No. 3-4 对四种送数方式的控制过程。其它各组都包含在这四种方式之中,读者将很容易分析。

当读数 A= $ D9A4=0 时,则本次向 PPU 送数的操作结束。关闭 PPU,返回调用程序。

本次共向 PPU 送入了七十六组数据。第一组向 PPU 的配色单元置入颜色代码,其分布情况见表 3-6;第二～第九组向背景 0 页的配色单元送入配色代码,配色分布见图 3-2;第十～第七十六组向背景 0 页的图象单元送入了画面数据,绘出了复位画面中背景页的部分。画面数据分布见图 3-3。

注——图 3-2、图 3-3 中的行地址加列地址为对应单元的地址。

表 3-6 复位画面中 PPU 配色单元数据表

PPU 地址	颜色代码	备注	PPU 地址	颜色代码	备注
3F00	0F	底背景配色	3F10	0F	底背景配色
3F01	30	BG 面 00 号配色	3F11	30	SP 面 00 号配色
3F02	00		3F12	27	
3F03	0F		3F13	0F	
3F04	0F	BG 面 01 号配色	3F14	0F	SP 面 01 号配色
3F05	30		3F15	30	
3F06	00		3F16	02	
3F07	0F		3F17	0F	
3F08	0F	BG 面 10 号配色	3F18	0F	SP 面 10 号配色
3F09	30		3F19	30	
3F0A	00		3F1A	16	
3F0B	0F		3F1B	0F	
3F0C	0F	BG 面 11 号配色	3F1C	0F	SP 面 11 号配色
3F0D	30		3F1D	30	
3F0E	00		3F1E	2A	
3F0F	0F		3F1F	0F	

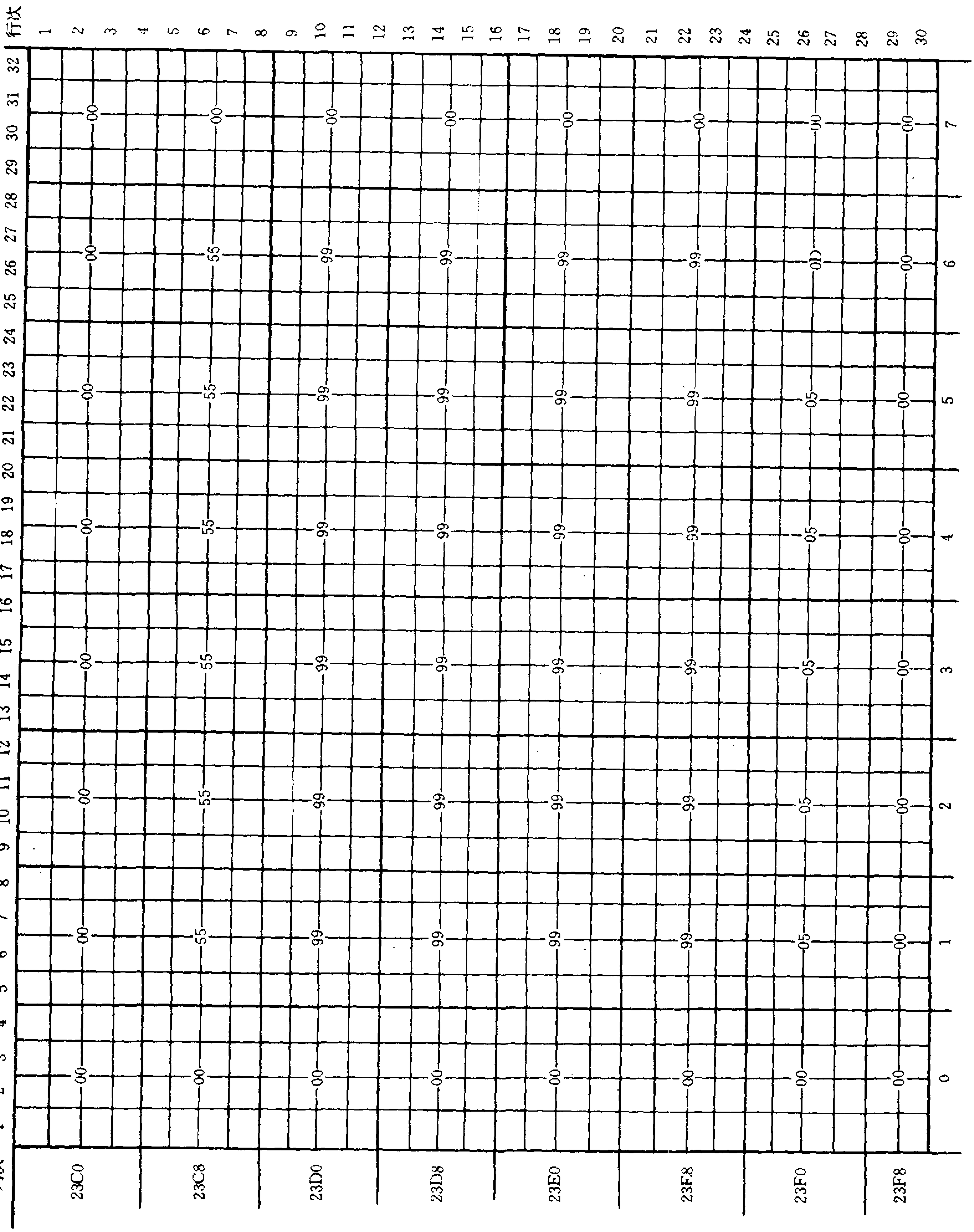


图 3-2 复位画面背景 0 页的配色分布

列次 / PPu 行地址	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	行次
2000	FE	16	2				16	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	FE	02				16	2	FE	1
2020	FE	16	2				16	1C	1C	1C	1C	FE	FE	FE	1C	1C	1C	1C	FE	FE	FE	1C	1C	1C	1C	02				16	2	FE	2
2040	FE	16	2				16	1C	1C	1C	1C	FE	FE	FE	1C	1C	1C	1C	FE	FE	FE	1C	1C	1C	1C	02				16	2	FE	3
2060	FE	16	2																											16	2	FE	4
2080	FE	16	2			9	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	11			16	2	FE	5
20A0	FE	1	0			A	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	12			1	0	1E	6
20C0	FE	1	0	3		7																					13		19	1	0	1E	7
20E0	FE	1	0	4	6	8																					11	17	1A	1	0	1E	8
2100	FE	1	0	5		7																					13		1B	1	0	1E	9
2120						7																					13			1	0	1E	10
2140	FE	1	0			7																					13			1	0	1E	11
2160	FE	1	0	3		7																					13		19	1	0	1E	12
2180	FE	1	0	4	6	8																					11	17	1A	1	0	1E	13
21A0	FE	1	0	5		7																					13		1B	1	0	1E	14
21C0	FE	1	0			B	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	15			1	0	1E	15
21E0	FE	1D	1F			C	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	18			1D	1F	FE	16
2200	FE	16	2			9	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	11			16	2	FE	17
2220	FE	1	0			A	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	12			1	0	1F	18
2240	FE	1	0	3		7																					13		19	1	0	1F	19
2260	FE	1	0	4	6	8																					14	17	1A	1	0	1F	20
2280	FE	1	0	5		7																					13		1B	1	0	1F	21
22A0	FE	1	0			B	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	15			1	0	1F	22
22C0	FE	1D	1F			C	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	18			1D	1F	FE	23
22E0	FE	16	2																											16	2	FE	24
2300	FE	16	1C			BE						BE							BE						BE					1C	2	FE	25
2320	FE	16	1C	B9	BC	BF	BC	BC	BC	B9	BC	BF	BC	BC	BC	B9	B9	BC	BF	BC	BC	BC	B9	BC	BF	BC	BC	BC	B9	1C	2	FE	26
2340	FE	16	1C	BA						BA						BA	BA							BA					BA	1C	2	FE	27
2360	FE	16	1C	BB	BD	BD	BD	BD	BD	BB	BD	BD	BD	BD	BD	BB	BB	BD	BD	BD	BD	BD	BD	BB	BD	BD	BD	BD	BB	1C	2	FE	28
2380	FE	16	1C																											1C	2	FE	29
23A0	FE	16	2																											16	2	FE	30
PPu 地址	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	

图3—3 复位画面中背景面图象单元数据

四、背景页绘图小结

背景页绘图程序为$EE09～$EE56。绘图方法为查表送数法。这种绘图法的优点是：

1、程序量小而功能强大

背景绘图程序仅占用H4D个内存单元，但可绘制出各种复杂的背景图形，而且设置配色和绘制图形共用一个程序。程序的使用效率较高。

2、数据量小而图形兼容

采用查表法绘图可把图形分区存放，以供各种画面对相同图形数据的共享。如《淘金者》游戏虽然有50版画面，但其基本图形类别量较小，因而图形的数据量是很小的。

3、编程工作量小而调试容易

查表法绘图，不论画面多么复杂、不论多少场画面，都不用修改控制程序。仅修改绘图数据就可妙手丹青、纵情挥洒。因此，查表法绘图编程简单易行、容易掌握，稍加练习就可照图直接输入数据。所以说，查表法绘图编程周期短、效率高、工作量小、容易调试，是所有绘图法中较优化的一种。

总之，查表绘图法是任天堂游戏中的主要绘图方法。希望读者能高度重视，力求全面领会掌握。

3·2·8 复位画面中卡通页图形的定义

复位画面中，各信号灯的闪烁（即信号灯的亮光）、阴体字"COMPUTER"、"OPERATOR"、黑体字"1、2、3、4"的显示，还有COMPUTER、OPERATOR栏下的8个白色方块，都是显示在卡通面上的。本节重点讨论这些卡通的定义。

一、阴体字"COMPUTER"显示卡通的定义

阴体字"COMPUTER"由6个卡通块组成，它们在卡通库（第1区）中的序号是HF4～HF9。

定义"COMPUTER"卡通块在第五次中断处理中完成。

在开机第五次中断中调$CE6E子程序，查表3－4得出程序的转移入口地址为$D515。在$D515程序中，通过调用$D68B子程序，完成了"COMPUTER"卡通块的定义。

定义过程如下：

1、由$D68B～$D694清0子程序，把$200～$2FF全部置为0。

2、令X＝6、调用$D644赋值子程序，对0页单元$00～$04单元赋值。

$D644是专为$00～$04单元赋值的子程序。赋值方法仍采用查表法；赋值目的是为了定义卡通块。

首先根据X寄存器的取值，从数据地址入口数据区$C4C7～$C4E0读出与X相对应的数据区入口地址存入$05、$06单元；然后根据$05、$06指出的地址取出5个数据赋于$00～$04单元。为方便使用，特列出子程序$D644中X寄存器值与数据入口地址对照情况于表3－7。

由表3－7可查出，当X＝6时，入口地址为$C4E9。因此，对$00～$04单元赋值如下：

$00＝68

$01＝24

$02＝F4

$03＝61

$04=00

表 3—7　$D644 中 X 寄存器值与 $00～$04 单元赋值对照表

X 寄存器取值	数据区地址（$5、$6 的值）	$0	$1	$2	$3	$4
0	C4E1	0C	3F	07	C5	0C
1	C4E1	0C	3F	07	C5	0C
2	C4E1	0C	3F	07	C5	0C
3	C4E1	0C	3F	07	C5	0C
4	C4E1	0C	3F	07	C5	0C
5	C4E5	0C	3F	0C	C5	68
6	C4E9	68	24	F4	61	00
7	C4EE	68	84	FA	61	18
8	C4F3	C6	20	50	03	14
9	C4F7	14	3F	50	03	44
A	C4FB	44	23	11	C5	67
B	C4FF	67	22	11	C5	44
C	C503	44	23	17	C5	14

第三、调 $ECBE 定义卡通块子程序，根据 $00～$04 单元的内容定义 6 个卡通块。$ECBE～$ED2C 是定义卡通块的专用子程序。调用前要对 $00 单元置入 SP 块的 X 坐标；对 $01 单元置入 SP 块的 Y 坐标；对 $02 单元置入第一个 SP 块的图库序号；对 $03 单元置入定义 SP 的数量和方式，其中高位表示定义 SP 块的数量、低位表示定义 SP 块的层数，因为现在定义的是 6 个横排的 SP 块，所以 $03=61；对 $04 单元置入 SP 块位于二页的地址低位。

第四、通过 $ECBE 子程序的处理，共定义 6 个显示 SP。数据如下：

$200=24　　$204=24
$201=F4　　$205=F5
$202=00　　$206=00
$203=68　　$207=70

$208=24　　$20C=24
$209=F6　　$20D=F7
$20A=00　　$20E=00
$20B=78　　$20F=80

$210=24　　$214=24
$211=F8　　$215=F9
$212=00　　$216=00
$213=88　　$217=90
$213=88　　$217=90

"COMPUTER"SP 块在背景 0 页的显示位置见图 3—4。

二、阴体字"OPERATOR"显示卡通的定义

阴体字"OPERATOR"也由 6 个 SP 块组成。它们位于图形库 I 区，序号是 HFA～HFF。定义"OPERATOR"SP 块也在第五次中断完成，控制程序为 $D69D～$D6A4。

它们的定义过程与"COMPUTER"基本相同，所不同的仅是它以 X=7 调用 $D644，

对＄00～＄04单元的赋值如下：

＄00＝68	＄01＝84
＄02＝FA	＄03＝61
＄04＝18	

然后调用＄ECBE子程序定义6个SP块。数据如下：

＄218＝84	＄21C＝84
＄219＝FA	＄21D＝FB
＄21A＝00	＄21E＝00
＄21B＝68	＄21F＝70
＄220＝84	＄224＝84
＄221＝FC	＄225＝FD
＄222＝00	＄226＝00
＄223＝78	＄227＝80
＄228＝84	＄22C＝84
＄229＝FE	＄22D＝FF
＄22A＝00	＄22E＝00
＄22B＝88	＄22F＝90

"OPERATOR"SP块在背景0页的显示位置见图3－4。

三、黑体字"1、2、3、4"显示卡通块的定义

黑体字"1、2、3、4"是四个提示栏上方的功能键提示"F"后的数字，由它们组成"F1、F2、F3、F4"四个功能键功能的提示。它们由四个SP块分别担任。这四个SP块的定义也在第五次中断完成，定义程序为＄D6A5～＄D6B0。

＄D6A5～＄D6B0是一个独立的子程序，直接调用可在屏幕下方显示黑体字"1、2、3、4"。

四个SP块的定义数据如下：

＄230＝C5	＄234＝C5
＄231＝F0	＄235＝F1
＄232＝00	＄236＝00
＄233＝38	＄237＝68
＄238＝C5	＄23C＝C5
＄239＝F2	＄23D＝F3
＄23A＝00	＄23E＝00
＄23B＝A0	＄23F＝D0

注—— 一、二、三项中的16个SP块可直接调用＄D68B而在屏幕上显示。（调用前，要保护好现场。）

四、用于信号灯闪烁和八个白色方块SP的定义

复位画面中共有34个信号灯和8个白色方块，在复位画面中定义了28个SP块控制它们的闪烁。其分工如下：

· 使用12个SP块决定上面两排24个灯闪烁，左、中、右各4个。

· 使用8个SP块担任两栏下框中的8个白色方块，每栏四个。

·使用 8 个 SP 块作用于左、右下角的 10 个信号灯，每边四个。

定义上述 SP 块的程序为 $D67F～$D68A。它是通过调用查表子程序 $D6B1，读出各 SP 块的数据置入 $240～$2AF 单元而实现的。各 SP 块的数据如下：

1、COMPUTER 栏中的四个闪烁方块

$240=73	$244=73
$241=DE	$245=DE
$242=02	$246=02
$243=B0	$247=B6
$248=73	$24C=73
$249=DE	$24D=DE
$24A=02	$24E=02
$24B=BC	$24F=C2

2、OPERATOR 栏中四个闪烁方块

$250=AB	$254=AB
$251=DE	$255=DE
$252=02	$256=02
$253=B0	$257=B6
$258=AB	$25C=AB
$259=DE	$25D=DE
$25A=02	$25E=02
$25B=BC	$25F=C2

3、作用于上两排信号灯的 12 个 SP 块

·作用于左边 8 个信号灯的 4 个 SP 块：

$260=07	$264=07
$261=DF	$265=DF
$262=F8	$266=F8
$263=F8	$267=F8
$268=0F	$26C=0F
$269=DF	$26D=DF
$26A=F8	$26E=F8
$26B=F8	$26F=F8

·作用于中间 8 个信号灯的 4 个 SP 块：

$270=07	$274=07
$271=DF	$275=DF
$272=F8	$276=F8
$273=F8	$277=F8
$278=0F	$27C=0F
$279=DF	$27D=DF
$27A=F8	$27E=F8

$ 27B=F8　　$ 27F=F8

·作用于右边 8 个信号灯的 4 个 SP 块：

$ 280=07　　$ 284=07
$ 281=DF　　$ 285=DF
$ 282=F8　　$ 286=F8
$ 283=F8　　$ 287=F8

$ 288=0F　　$ 28C=0F
$ 289=DF　　$ 28D=DF
$ 28A=F8　　$ 28E=F8
$ 28B=F8　　$ 28F=F8

可以看出，最初定义时这 12 个 SP 块中每 6 个为相同的坐标。这要在以后的中断处理中才能令其作用在各自的信号灯上。采用这种定义方法的优点是减小程序量和压缩数据区内存。

4、作用于左下角 5 个信号灯的 4 个 SP 块：

$ 290=F8　　$ 294=F8
$ 291=DF　　$ 295=DF
$ 292=F8　　$ 296=F8
$ 293=10　　$ 297=10

$ 298=F8　　$ 29C=F8
$ 299=DF　　$ 29D=DF
$ 29A=F8　　$ 29E=F8
$ 29B=10　　$ 29F=10

这四个卡通块在初定义时也挤在一起，原因同 3。

5、作用于右下角 5 个信号灯的 4 个 SP 块：

$ 2A0=F8　　$ 2A4=F8
$ 2A1=DF　　$ 2A5=DF
$ 2A2=F8　　$ 2A6=F8
$ 2A3=E8　　$ 2A7=E8

$ 2A8=F8　　$ 2AC=F8
$ 2A9=DF　　$ 2AD=DF
$ 2AA=F8　　$ 2AE=F8
$ 2AB=E8　　$ 2AF=E8

初定义时这四个 SP 块也挤在一起。

各 SP 块在背景 0 页的对应位置见图 3－4。

五、SP 块定义过程小结

$ ECBE 是定义 SP 块的专用子程序，可供我们在设计机器语言程序中调用。它也是一个重要子程序，但在调用该子程序时要注意以下两点：

1、该子程序工作中要占用 $ 00～$ 09 单元和 A、X、Y 寄存器，因此在调用前要做好上述现场的保护。

2、调用前要把定义 SP 的参数存入 $ 00～$ 04 单元中。各单元的内容如下：

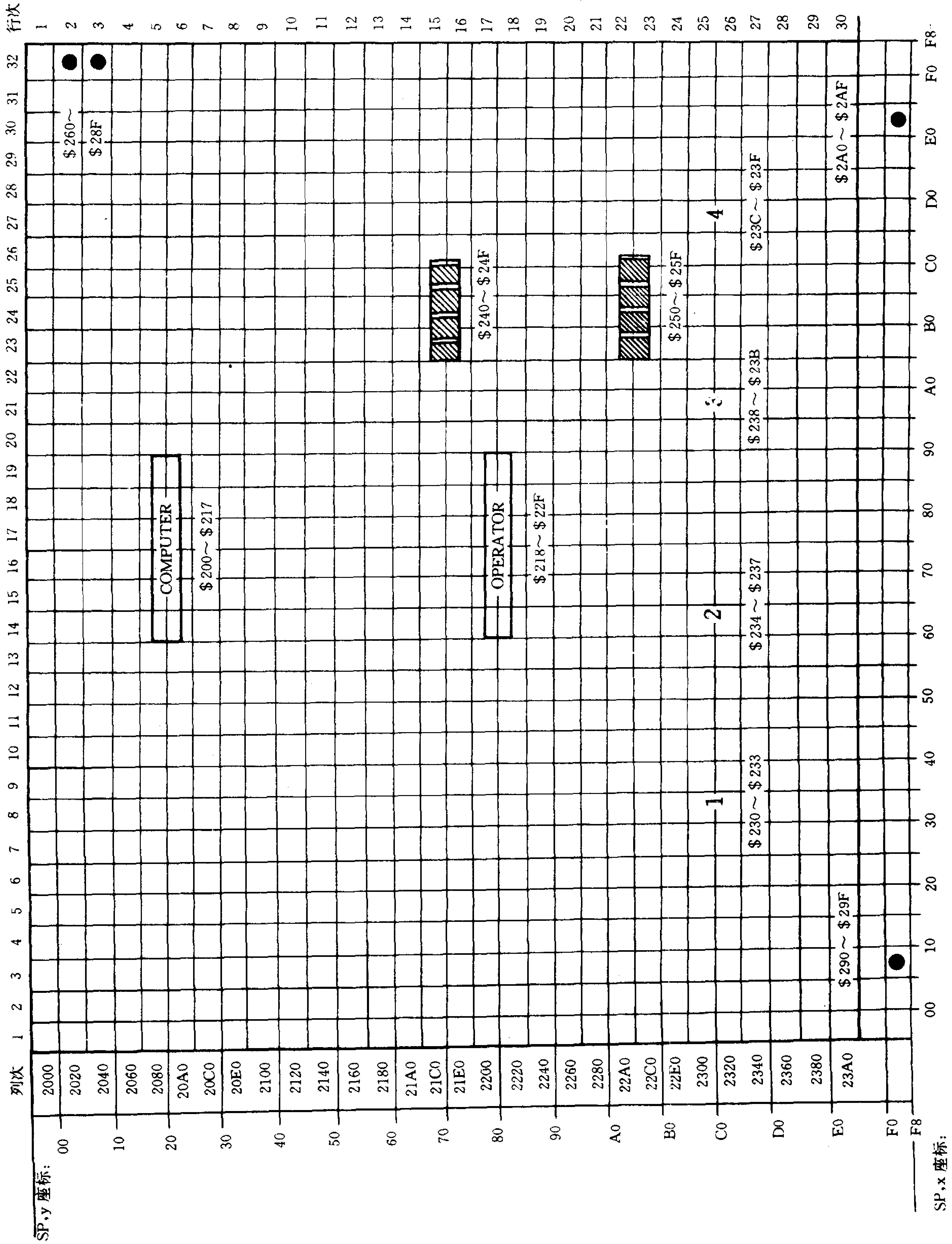


图 3－4　复位画面中SP块初定义时在背景面中的位置示意图

$00:SP 块的 X 坐标。

$01:SP 块的 Y 坐标。

$02:SP 块的序号。

$03:一次定义 SP 块的数量和 SP 块的排列方式。

$04:定义的 SP 块位于二页的地址低位。

3·2·9 功能键的定义

F1～F8 八个功能键的定义在第四次中断完成。在执行第四次中断时,通过调用 $CE6E 设置程序转移入口为 $F4B1。

$F4B1 是定义功能键的专用子程序。定义过程仍然采用查表法。控制程序位于 $F4B1～$F4C5;功能键数据区位于 $EED3～$EEFA,共有五组功能键数据。数据组的选定由调用该子程序时 X 寄存器的取值决定。它们的对应关系见表 3－8。

表 3－8 $F4B1 中 X 寄存器值与功能键定义内容对照表

X	F1	F2	F3	F4	F5	F6	F7	F8	备　注
0	C1	C2	C3	C4	C5	C6	C7	C8	对话板
1	2B	2D	B5	B6	28	29	2E	3D	计算板
2	63	64	65	66	67	61	62	C8	音乐板
3	02	06	0E	0F	10	11	17	18	未用
4	43	44	45	46	47	41	42	24	音乐板

复位处理中调用 $F4B1 时,X＝0,所以把八个功能键定义为 HC1～HC8。

$F4B1 子程序也可在编程中直接调用。调用前只要对 X 寄存器置入相应的数值(0～4),直接调 $F4B3 就可以按表 3－8 的数值定义功能键。$F4B1 工作中要使用 A、X、Y 寄存器和 $60 单元,因此要做好现场保护。

3·2·10 复位画面中 OPERATOR 栏的操作

在复位画面中,OPERATOR 栏呈天蓝色。从开机后到按任一键前一直保持这种状态。它也是复位画面绘制工作中的一部分。把 OPERATOR 栏置为天蓝色的过程是向栏中写入背景图形 HFF。写入操作要经过三次中断操作才能完成。每次中断写一行,分别在第五～第七次中断写完。

置色操作控制程序为 $D5F9～$D61B,由 $B7 控制置色选行。$B7 是对 COMPUTER、OPERATOR 两栏各行操作的控制器。根据 $B7 的取值可以查出各行行首的 PPU 地址,然后对选定行进行置色、显示字符、置入空格(行清除)等操作。

表 3－9 对话板中 $B7 取值与 PPU 地址对照表

$ B7	PPU 地址 $B9、$BA	备　注	$ B7	PPU 地址 $B9、$BA	备　注
1	20C6	电脑栏第一行	7	2186	电脑栏第七行
2	20E6	电脑栏第二行	8	21A6	电脑栏第八行
3	2106	电脑栏第三行	9	2246	操作栏第一行
4	2126	电脑栏第四行	A	2266	操作栏第二行
5	2146	电脑栏第五行	B	2286	操作栏第三行
6	2166	电脑栏第六行			

在对话板的工作中，$B7 是一个重要的 0 页单元。它在对话字符的送显、清除中使用极为频繁。现将 $B7 取值与各行对应的 PPU 首地址汇集列于表 3－9，以供查用。

在置色操作中要反复调用一个重要子程序 $D3FE。$D3FE 也是赋值子程序。其作用是：根据 $B7 的取值，查出被操作行的 PPU 地址存入 $B9、$BA 中（低位在前）；然后通过 $00～$05 单元对三页置入向 PPU 写入的送数指南和数据，（以便在下一次中断中，由 $CDD3 调用 $EE43 子程序把数据送入 PPU。）数据结构与 $D7F4 数据库的数据相同。

向 OPERATOR 栏第一行置色时，$B7＝9，置入三页各单元的数据如下：

$300＝04
$301＝22
$302＝46
$303＝54
$304＝FF
$305＝00

它们表示：从 PPU 的 $2246 地址开始，横向置入 H14 个 HFF。$305＝00 为向 PPU 写入结束标志。

在第六次中断，$B7＝HA，置入三页的数据如下：

$300＝04
$301＝22
$302＝66
$303＝54
$304＝FF
$305＝00

在第七次中断，$B7＝HB，置入三页的数据如下：

$300＝04
$301＝22
$302＝86
$303＝54
$304＝FF
$305＝00

通过以上三次中断，完成了把 OPERATOR 栏中的三行置为蓝色的操作。

在对 OPERATOR 栏操作中，经常调用的重要子程序介绍：

1、赋值子程序 $D3FE

$D3FE 子程序位于 ROM 区 $D3FE～$D420。它的主要作用是：在对 COMPUTER、OPERATOR 两栏的清除操作中对有关单元赋值。在对话工作中，COMPUTER 栏要不停地显示提问和叙述语句；OPERATOR 栏则要回答操作者的各种陈述。因此，这两工作栏的写入和清除操作是极为频繁的。为简化叙述，在今后的讨论中我们把 COMPUTER、OPERATOR 两栏的清除过程简称为清两栏。

2、向三页送数子程序 $ED65

$ED65 子程序位于 ROM 区 $ED65～$EDD1。它是一个独立的子程序，程序中含有对 X、Y 寄存器的现场保护操作。它的主要作用是：根据 $00、$03 单元的内容，把有关数据送入

三页，以设置＄EE43送数子程序的工作数据模式，向PPU传送数据。我们可以直接调用该子程序。但在调用前要对＄00、＄03单元预置数值。各单元数据含义如下：

·＄00:PPU的地址低位。

·＄01:PPU的地址高位。

·＄02:取数地址低位。

·＄03:取数地址高位。

对＄02、＄03单元指示的数据区中的数据要求如下：

·第一个数据必须是送数指南中的S3。

·其后紧挨的是送入PPU的数据。

经＄ED65子程序处理后，将在＄300单元指出占用三页的单元数；以后各单元的内容与＄D7F4数据区的结构相同；数据尾后的一个单元置0。

注意：＄ED65子程序，每次最多可送H3F个数据，对应于背景页的两行。

当OPERATOR栏全部置为蓝色后，则令＄B6＝3，置OPERATOR栏操作完毕标志。然后在下一次中断调＄D7D3子程序令＄3F07＝21；再调＄D63C子程序令＄3F0C～＄3F0F＝0F，以便向COMPUTER栏置入图形块时全部显示灰色。这两个子程序都较简单，读者可对照表3－7自行分析。

3·2·11 复位画面中提示栏的操作

复位画面的最下一行是提示栏。提示栏共有四块、横向排列，分别为：

·F1:"OK"栏

·F2:"NO"栏

·F3:"HELLO"栏

·F4:"END"栏

提示栏的作用是：提示操作者选择某一种回答时所按的功能键。

在复位画面中提示栏同OPERATOR栏一样，也被置为天蓝色。提示栏的置色操作分四次完成，每次中断置一栏，置色顺序为"END"栏、"HELLO"栏、"NO"栏、"OK"栏。

提示栏的置色控制程序为＄D5EB、由＄C7选定对哪一栏置色。＄C7的取值与提示栏的对应关系见表3－10。

表3－10 ＄D5EB中＄C7取值与各提示栏数据对照表

＄C7	PPU的起始地址	＄00	＄01	＄02	＄03	提示栏类别
0	2344	44	23	17	C5	OK
1	234A	4A	23	17	C5	NO
2	2351	51	23	17	C5	HELLO
3	2357	57	23	17	C5	END

提示栏的置色处理子程序为＄D47D～＄D48E。它首先从提示栏的PPU地址数据区＄CA64～＄CA67取出PPU的地址低位；然后调＄D644赋值子程序对＄01～＄03单元赋值；最后向＄00单元置入PPU地址低位，转＄ED65把置色数据送入三页。各栏的置色数据如下：

1、"END"栏

$ 301＝23　　　　$ 302＝57

$ 303＝05　　　　$ 304＝FF

$ 305＝FF　　　　$ 306＝FF

$ 307＝FF　　　　$ 308＝FF

$ 309＝0

2、"HELLO"栏

$ 301＝23　　　　$ 302＝51

$ 303～$ 309 同"END"

3、"NO"栏

$ 301＝23　　　　$ 302＝4A

$ 303～$ 309 同"END"

4、"OK"栏

$ 301＝23　　　　$ 302＝44

$ 303～309 同"END"。

注——对提示栏置色控制程序 $ D5EB 可作为子程序直接调用。调用时首先保存 A、X、Y 寄存器、$ 00～$ 09 单元的内容；然后对 $ C7 置入恰当的数值，就可在屏幕下方的相应位置显示五块相连的蓝色长条；调用后要及时恢复现场。

3·2·12　COMPUTER 栏三基色块的绘制过程

在复位画面中，COMPUTER 栏有 20 列×8 行的空间，显示着 160 个红、绿、蓝三基色方块。这些彩色方块的绘制在第十四次至第十八次中断中完成。绘制方法是：把 COMPUTEK 栏以每 4 列×8 行为单位分成五大块，每一次中断绘制一块，分五次完成。

绘制 COMPUTER 栏的控制程序共分为两段。第一段为 $ D543～$ D569，完成每一块 4 列×8 行图块的图形号的选取，选取的图形序号存入 $ 350～$ 36F 单元。同时检查绘图进度，若整栏绘制完毕则控制转出绘图程序。

第二段为 $ D61C～$ D634，完成把存入 $ 350～$ 36F 中的图形序号、配置送数指针后存入 $ 301 开始的各单元。PPU 地址低位存 $ B7 单元，每绘完一块令 $ B7 加一指向下一块的 PPU 首址低位。

下面以绘制 COMPUTER 栏第一块图形为例介绍绘制过程。

1、执行 $ D543。首先向 $ B7 置入第一块图形的 PPU 首址低位，令 $ B7＝C6；然后令 $ 51＝C6 调 $ EDE8 随机数发生子程序，取 $ 52 AND 3 的数值为图形序号选择指针送入 X 寄存器变址，从 $ C525 数据区选出与之对应的图形序号送入 $ 350 开始的各单元。当 $ 350～$ 36F 置满后则转入第二阶段。X 寄存器值与图形序号的对应关系见表 3－11。

表 3－11　$ D543 中 X 寄存器值与图形序号对应关系对照表

X 寄存器值	图形序号地址	图形序号	图形色彩
0	C525	FF	红
1	C526	FE	蓝
2	C527	20	黑（空格为黑）
3	C528	FD	绿

2、执行 $ D61C。首先令 $ 350＝H84，设置向三页送数方式；然后令 X＝8，调 $ D644 赋值

程序对＄00～＄04单元赋值，查表3－7，当X＝8时有：＄01＝H20、＄02＝H50、＄03＝3；最后把＄B7的PPU地址低位送入＄00，调向三页送数子程序＄ED65，把送数指南和图形序号送入＄301开始的各单元。

第一块图块的数据如下：

＄301＝20　　　＄308＝20
＄302＝C6　　　＄309＝E6
＄303＝04　　　＄30A＝04
＄304～＄307为4个图形的序号
＄30B～＄30E为4个图形的序号

＄30F＝21　　　＄316＝21
＄310＝06　　　＄317＝26
＄311＝04　　　＄318＝04
＄312～＄315为4个图形的序号
＄319～＄31C为4个图形的序号

＄31D＝21　　　＄324＝21
＄31E＝46　　　＄325＝66
＄31F＝04　　　＄326＝04
＄320～＄323为4个图形的序号
＄327～＄32A为4个图形的序号

＄32B＝21　　　＄332＝21
＄32C＝86　　　＄333＝A6
＄32D＝04　　　＄334＝04
＄32E～＄331为4个图形的序号
＄335～＄338为4个图形的序号
＄339＝0　　　结束标志

3、送数结束后，令＄B7＋4指向下一块图形的PPU首址低位。

以后四块图形的绘制过程同上(分析从略)。

3·2·13　各信号灯的闪烁控制

在复位画面中，34个信号灯不停地进行着无规则的闪烁和无规律的色彩变化，从而丰富了画面的绚丽色彩、产生了迷人的动感效果。

信号灯变化的实质就是随机改变信号灯的配色和在信号灯上叠加亮点SP块，并随机控制SP块对某一信号灯的叠加周期，从而仅使用20个卡通块就产生出全屏34个信号灯随机闪烁的奇妙效果。(《花式台球》游戏中就采用了类似的控制方式。)

一、控制程序的特点和作用

信号灯闪烁控制程序位于ROM区＄D709～＄D7D2。这是一个较复杂的子程序，按功能大体上可分为五部分：

1、子程序＄D77B，用于调整OPERATOR栏四个方块的显示状态。

2、子程序＄D659，用于调整PPU＄3F03背景配色单元的颜色代码。

3、$D712～$D72D,用于调整 PPU $3F14～$3F17SP 配色单元的颜色代码。颜色代码数据取自 $52～$55 单元中的随机数值;调整间隔由中断计数器 $3B 控制;调整条件是 $3B AND 3=1。

4、$D72E～$D73E,用于调整 PPU $3F0B 配色单元的颜色代码。颜色代码数据取自 $51 单元中的随机数值;改变条件是 $B0～$B3 都为 1。

5、$D73F～$D76C,用于调整作用于 34 个信号灯的 20 个 SP 块的状态及坐标,调整条件也是 $3B AND 3=1。

二、控制过程

开机后,第一次调整 SP 块的操作是在第九次中断完成的。调整过程分五步进行。

1、OPERATOR 栏四个方块闪烁的控制

OPERATOR 栏中四个白色方块闪烁的控制由 $D77B 子程序完成。当 OPERATOR 栏没有操作时,$24=0,则四个 SP 块的状态有

$252=$256=$25A=$25E=02

故它们保持常态,呈白色。

当 OPERATOR 栏操作后,当中断计数器 $3B 的低位为 2、3、6、7、H0A、H0B、H0E、H0F 时则根据 $51 单元的数值调整四个 SP 块的状态,使它们的状态从 0、1、2、3 中随机选取,令其无规则闪烁。

2、调整 $3F03 的颜色代码

调整 PPU $3F03 单元中的颜色代码,目的是调整背景配色板中的颜色组合,改变各信号灯的色彩。$D569 就是调整 $3F03 内容的专用子程序。工作过程如下:

当 $22=0 时,每当 $3B 的低位为 1、9、H0D 时则调整 $3F03 中的颜色代码,颜色数据由信号灯颜色控制器 $B8 决定。$B8 取值与颜色代码的对应关系见表 3－12。

表 3－12 $D659 中 $B8 取值与 $3F03 颜色代码的对应关系

B8	3F03	颜色种类	B8	3F03	颜色种类
0	16	亮洋红	4	0F	黑
1	6	大红	5	8	黄
2	7	柿黄	6	7	柿黄
3	8	黄	7	6	大红

当 $3F03 的颜色连续调整七次后,则置调整暂停延时器 $22=30,暂停 0·94 秒后再恢复调整。反映在画面上就是各信号灯闪烁 7 次后,暂停一会再继续闪烁。

3、调整 SP 块配色

调整各 SP 块配色的处理程序为 $D712～$D72D,执行条件是每隔四个中断周期调整一次。调整过程是:首先取出 $52～$55 四个单元的随机数送入 $351～$354;然后令 $350=H14、X=9,调 $D644 子程序,由表 3－7 查出 X=9 时的对应值送入 $00～$03 单元;它们分别为:

$00=14　　$01=3F

$02=50　　$03=03

$04=44

最后转 $ED65 把 PPU 地址及数据送入 $301 开始的各单元,以便在下一次中断期间置入 $3F14～$3F17,随机地改变卡通配色板的 1 号配色代码的颜色组合。

4、调整＄3F0B颜色代码

调整＄3F0B颜色代码的处理程序在＄D72E～＄D73E。执行这一程序的条件是：＄B0～＄B3同时为1，即COMPUTER栏的彩色方块置入完毕后才执行；也是每隔80毫秒(四个中断周期)调整一次；调整数据取自＄51单元的随机数；目的是改变画面两边支柱图形的色彩。

5、调整SP块的坐标及状态

调整SP块的显示坐标和状态的控制程序在＄D73F～＄D76C。它主要作用于34个信号灯，共控制20个SP块。这20个SP块共分四类进行操作：

第一类：作用于上排左端信号灯的四个SP块。

第二类：作用于上排中间信号灯的四个SP块。

第三类：作用于上排右端信号灯的四个SP块。

第四类：作用于左、右下端信号灯的八个SP块。

下面分别讨论程序的处理过程：

・以X寄存器为变址指针，令X＝3，先调整左、右下端的八个SP块。

首先由＄D73F～＄D75C处理，调整各SP块的状态；然后由＄D75D～＄D768处理调整各SP块的坐标。当执行第九次中断时，＄3B＝9，有＄3B AND 3＝1。假定我们取＄51～＄58单元的第100次随机数，则调整后8个SP块的数据如下：

＄290＝C7	＄2A0＝BF
＄292＝01	＄2A2＝02
＄294＝BF	＄2A4＝C7
＄296＝00	＄2A6＝02
＄298＝CF	＄2A8＝BF
＄29A＝02	＄2AA＝00
＄29C＝D7	＄2AC＝CF
＄29E＝03	＄2AE＝03

由以上数据可见，左下角的4个SP块已与左下角的第二～第五个信号灯重合，即令这四个信号灯发出不同色彩的亮光(由于它们的状态单元分别对应了0、1、2、3四个配色代码。)；而右下角的4个SP块则与右下角的第一～第三3个信号灯重合，令这三个信号灯发出不同色彩的亮光。

・调整第三类SP块

第四类SP块调整完毕，令X＝X－1，则调整第三类SP块——上排右端的四个SP块。调整过程是先调整SP块的状态、后调整SP块的X坐标。调整后的数据如下：

＄282＝02	＄28A＝00
＄283＝A8	＄28B＝A8
＄286＝02	＄28E＝01
＄287＝B0	＄28F＝B8

可见，通过调整X坐标，这四个SP块已分别与右端上排的第一、第二列信号灯和下排的第二、三列信号灯相重合。

・调整第二类SP块

调整完第三类SP块后，令X＝X－1＝1，调整第二类。调整过程同第三类。调整后的数据如下：

$ 272=02　　$ 27A=00

$ 273=70　　$ 27B=70

$ 276=02　　$ 27E=01

$ 277=78　　$ 27F=80

第二类 SP 块分别与中间上排的第一、二列和下排的第二、三列信号灯相重合。

· 最后 X=0,则调整第一类 SP 块。数据如下：

$ 262=02　　$ 26A=02

$ 263=38　　$ 26B=38

$ 266=02　　$ 26E=02

$ 267=40　　$ 26F=48

这四个 SP 块分别与左端上排的第一、二列和下排的第二、三列信号灯重合。通过九次中断处理后，作用于信号灯的 20 个 SP 块已基本与信号灯重合。以后在 $ 3B=H0D、$ 3B=H11……等中断计数器 $ 3B 的 D1、D0 位为"01"时，将通过 $ 51～$ 54 单元的控制，调整各 SP 块随机叠加到各信号灯上，从而产生无规则闪烁的画面。

为便于讨论，特把 X 寄存器的取值与操作 SP 块种类的有关数据列于表 3－13。

表 3－13　$ D73F 中 X 取值与调整 SP 块类别对照表

X 寄存器中的数值	SP 块的工作区域	SP 块的类别	某类 SP 块的状态数据首址低位	某类 SP 块的坐标数据首址低位	某类 SP 块的坐标基本量	一次调整 SP 块数量
			$ D757　用	$ D763　用	$ D7B2　用	$ D792
0	上左 8 灯	一	62	63(X)	38	0C
1	上中 8 灯	二	72	73(X)	70	0C
2	上右 8 灯	三	82	83(X)	A8	0C
3	左下、右下 10 灯	四	92	90(Y)	BF	1C

注——在一次调整 SP 块数量一栏中的数据是地址增量值。实际的 SP 块数应用该栏的数值除以 4 后再加 1。如：

X=0 时，调整 HC÷4+1=4 个 SP 块

X=3 时，调整 H1C÷4+1=8 个 SP 块

3·2·14　等待按键状态的建立

复位画面绘制完成后则进入等待按键状态。这时各信号灯不停地变换着色彩和亮度；COMPUTER 栏的三基色彩块不停地移动位置；背景忽明忽暗；扬声器发出的声响规律为：开始是连续的嘟嘟声、随后是无规则的滴哒声、再后又是连续的变调嘟嘟声、最后则无声。延时一段时间后又重复以上发声。以上过程一直持续到操作者按任一键后结束。

等待按键状态的建立过程如下：

1、从第十四次中断开始，经过五个中断周期把 COMPUTER 栏全部置满彩色方块后转 $ D56A，令 $ B6=5 置 COMPUTER 栏绘图结束标志。

2、在第十九次中断期间，由 $ B6=5 转 $ D589 配色处理子程序。

$ D589 是专为 COMPUTER 栏显示三基色块而设置的配色程序。由 X=4 可从表 3－7 中查出对应的数据为：

$ 00=0C　　$ 01=3F

$02=07　　　　$03=C5

然后转$ED65子程序把令

$3F0C=0F　　　　$3F0D=16

$3F0E=02　　　　$3F0F=2A

的送数指南和数据送入三页，从而使置入COMPUTER栏的四种色块(H20、HFD、HFE、HFF)显示出黑、红、绿、蓝三基色。最后置配色结束标志$B6=6。

3、在第二十次中断时，由$B6=6转$D591，正式进入等待按键状态。

$D591是复位画面等待按键处理程序。每执行一次中断检查一次$37单元的数值。若$37≠0则表示有按键，转入有按键处理程序，预备进入对话板；若$37=0则转$D5A0进行色块组合和发声处理。

色块组合和发声控制子程序$D5A0～$D5AF是等待按键状态的重要工作程序。由它完成COMPUTER栏三基色块的无规则组合和间歇发声控制。

(1)色块组合处理

COMPUTER栏色块组合的随机性由$53单元决定。它以$53 AND 7的数值变址，选定COMPUTER栏改变色块组合的PPU地址低位。色块组合的变化仍然使用向COMPUTER栏置图子程序$D543，每次仍然改变4列×8行32个图形块，图形块的种类仍然通过$52单元的数值随机决定。所不同的仅是：每次改变4列×8行图形块的起始地址随机选定，从而决定了COMPUTER栏三基色块变化的随机性。其改变情况与$53单元的取值对应关系见表3—14。

表3—14　复位画面中COMPUTER栏彩块变化表

$53 AND 7	改变COMPUTER栏彩块的起始地址低位	改变COMPUTER栏彩块的起始列次
0	C6	第1列
1	CA	第5列
2	CE	第9列
3	D2	第13列
4	D6	第17列
5	CC	第21列
6	D0	第25列
7	D4	第29列

(2)发声的控制

发生控制由子程序$D5B9～$D5D7完成。控制过程如下：

发声开始时，$2C=0。$2C减一后，$2C=HFF，发声计时开始。转$D5D1、令$FC=70，发出连续的嘟嘟声。

当经过300个中断周期、即6秒钟后，$2C=HF0、则令$FC=H10，开始发无规则的滴哒声。

当滴哒声持续12.8秒(640个中断周期)后，$2C=HD0，则又令$FC=H70，发出变调的嘟嘟声。

嘟嘟声发出3.2秒(160个中断周期)后，$2C=HC8，则令$FC=0，停止发声。

当停止发声80秒后，$2C=0，又重新启动发声，重复以上过程。

发声过程见表3—15。

表 3－15 复位画面中发声阶段对照表

发声阶段	$2C 的取值范围	$FC	持 续 时 间	声 调 种 类
1	F0 ～ FF	70	6.0 秒	连续的嘟嘟声
2	D0 ～ EF	10	12.8 秒	不规则的滴哒声
3	C8 ～ CF	70	3.2 秒	连续的嘟嘟声
4	00 ～ C7	0	80.0 秒	无声

关于发声的工作原理和过程，我们将专题讨论，此处从略。

4、在等待按键状态下的每一次中断处理，都要调用 $D709 子程序，根据中断计数器 $3B 的数值，改变各 SP 块的状态、坐标和 $3F03、$3F0B、$3F14～$3F17 的颜色代码。改变过程同以上所述(从略)。

3·2·15 复位工作过程小结

1、系统的复位操作全部在中断处理中完成。开机或按复位键后，自动执行 $C400 程序，设置中断入口为 $CDA7。

2、复位画面的绘制由绘画进程控制器 $B6 控制。$B6 的取值与绘画进程对应如下：

$B6＝1：绘制复位画面。

$B6＝2：把 OPERATOR 栏置为蓝色。

$B6＝3：把提示栏置为蓝色。

$B6＝4：向 COMPUTER 栏置入 HFD、HFE、HFF、H20 四种图块。

$B6＝5：向 PPU 配色单元(背景配色板的 3 号代码)置入配色，使 COMPUTER 栏的图形显示红、绿、蓝、黑四种颜色。

$B6＝6：进入等待按键状态。

3、在绘制复位画面的同时，由中断计数器 $3B 控制定义了 44 个 SP 块。它们的地址和作用如下：

- 定义 $200～$217 六个 SP 块显示 COMPUTER
- 定义 $218～$22F 六个 SP 块显示 OPERATOR
- 定义 $230～$23F 四个 SP 块显示"1、2、3、4"
- 定义 $240～$25F 八个 SP 块显示两栏中的八个白色方块
- 定义 $260～$2AF 二十个 SP 块作用于各信号灯

4、重要子程序

- $EE09：向 PPU 送数子程序
- $ED65：向三页送数子程序
- $ECBE：定义 SP 块子程序
- $EDE8：随机数产生子程序
- $EDD2：计数器递减子程序
- $F4B1：定义功能键子程序。

3·4 对话板的工作过程

对话板是进入音乐、计算、打字各板和 F BASIC 系统的控制中枢。通过对话板工作过程的

分析，我们可以从中了解掌握送显字符和部分清屏的操作方法，进一步加深对查表技术的认识。

3·3·1 对话板的进入过程

在复位画面的等待状态按任一键后，由＄D591程序读出键码，令＄B6加一，置有按键标志。随后令＄FC＝0以关闭无规则"滴哒"声；令＄FE＝4，发出进入对话板的"嘟噜"声。然后进入对话板处理。

1、调整配色

调＄D635配色子程序，令＄3F07＝H0F、＄3F0C～＄3F0F＝H0F，把COMPUTER栏置为黑色背景。同时令＄60＝0，为下一次中断转移程序入口的重新设置做好准备。

2、清除两栏

在按键后的第二次中断处理中，令＄60＝6，设置了新的程序转移入口为＄D2A4。

＄D2A4是对话板的主要工作程序，两栏的各项操作都由该段程序控制完成。进入对话板时，先进行清两栏操作。

清两栏的处理程序在＄D3FE～＄D420，这是一个专用部分清屏子程序。清两栏操作共分11步完成，由＄B7控制每次中断清除一行。清除顺序是：从COMPUTER栏的第一行开始到OPERATOR栏的第三行结束。清栏开始时＄B7＝1，每清一行＄B7加一，直到＄B7＝H0C时则转清栏结束处理程序＄D421，令＄C6＝1以设置清栏结束标志，令＄3F0D＝H30以置两栏字符为黑底白字配色。

3、COMPUTER栏送显字符

送显字符的控制程序也在＄D2A4。当清两栏结束后，因＄C6＝1则转入＄D2C4。

＄D2C4是送显字符的控制程序，在该程序中由＄C2选定送显字符处理程序的种类。因为在COMPUTER栏的字符送显中有两类字符：一类是叙述语句字符，该类包括：微机系统的介绍、生产厂家的通信地址、向操作者的问候、祝愿等等；第二类是系统名词字符，它们包括各板名称(MUSIC、CALCULATOR、MESSAGE、F BASIC等)、各板名称的简写MUS·、CAL·、MES·、等)、选项命令(OK、NO、HELLO、END)等等。这两类字符的数据区不同(第一类位于＄C5A1～＄CA36、第二类位于＄CAB7～＄CB27)、送显字符的控制程序也不同。因此由＄C2控制选择送显程序。当送显叙述字符时＄C2＝0；送显名词字符时＄C2＝1。

由于进入对话板时的送显是叙述字符，＄C2＝0，所以转＄D2DA叙述字符送显子程序。

＄D2DA～＄D326是送显字符操作主程序。它通过调用＄D3E0取数子程序，首先由＄C3决定数据区首址；然后由＄C5变址取出送显字符数据。＄C3的取值在＄CE6E程序中与＄60同时决定，取值范围为0～H39。为方便使用，特将＄C3对应的送显字符汇集于表3－16。

表3－16 ＄C3单元的取值与送显字符对照表

＄C3	数据区首地址	与＄C3对应的送显字符内容
00	C7A1	I'M CNEDC FAMILY COMPUTER SYSTEM
01	C5B5	NOW WE START
02	C5C2	OK
03	C5C9	YES
04	C5CD	OK!
05	C5D4	NOW WE START!

$C3	数据区首地址	与 $C3 对应的送显字符内容
06	C5E2	WHO ARE YOU? //INPUT NAME
07	C5FC	KEY AGAIN!
08	C607	"C2"
09	C613	NOW WE...
0A	C622	BYEBYE
0B	C629	CASSETTE'S SWITCH//OFF PLEASE!
0C	C647	"C2"
0D	C647	"C2"// NOW FINISHED
0E	C724	空
0F	C680	HELLO
10	C686	NEXT SELECT.
11	C694	OK! SORRY!
14	C945	CNEDC SHENZHEN CO. TEL: 361105 365402 CA-BLE:4703
15	C6CA	"C0" HAVE A REST?
16	C6DB	TIME OVER
19	C6F8	HAVE A REST. //OK
2D	C6AB	NOW WILL BE //CONTINUE !
2F	C97C	OK! //"C0"//IS YOU
30	C98A	"C0"//DO YOUR HOMEWORK
34	C9B7	WILL WORK AGAIN
36	C9CF	C0!
37	C9D2	C0//HOW ARE YOU
38	C9E1	CNEDC SHENZHEN CO. IS GLAD TO SERVE FOR YOU. //THANKS!
39	C99D	PRESSING RESET COMPUTER WILL WORK A-GAIN

注——表中"//"为换行符号,要另起一行显示。

表 2-16 是对话板工作中的重要数据表,从表中的送显字符可以看出,它包括了对话板工作的全部字符。

在送显字符过程中,每一次中断只送显一个字符。同时令 $FF=2 以发出送显字符时的"滴滴"声。

在送显字符的过程中,若读到"0D"时则转 $D35A 换行处理程序使输出显示换一行;若读到"C0"时则转 $D37E 送显操作者的姓名;若读到"C2"时则转 $D392 送显指定的系统名词;若读到"FF"时则表示本次送显结束,转 $D327 进行送显结束处理。并令 $FE=2,发出本次送显结束的"叽哩咕噜"声响。同时,令 $2A=58,启动显示延时器,开始送显字符的显示延时,直到 $2A=0、即延时 1.7 秒后,清栏,再显示下一组字符。

4、对话板建立过程中 COMPUTER 栏的送显内容如下:

送显"I'M CNEDC FAMILY

COMPUTER SYSTEM"

每送完一次令四个小方块闪烁。

延时 1.7 秒后清屏、送显

"NOW WE START"

延时 1.7 秒后清屏、送显

"WHO ARE YOU?

INPUT NAME"

5、定义光标

在执行按键后第 355 次中断时，调 $CE6E、令 $60=H10，查表 3-4 得出程序的转移入口地址为 $D0E8。经调用 $D05D 子程序后定义 OPERATOR 栏的光标。光标的初始数据为：

$2C0=97　　　$2C1=DD

$2C2=00　　　$2C3=38

置长延时计数器 $2D=HFF，进入操作者输入姓名的等待状态。并令 $FC=H68，发出轻微的"哆哆"声，等待操作者输入姓名。

等待延时时间为 102 秒，若超过 102 秒没有输入姓名按回车键，则自动返回复位画面的等待按键状态。

3·3·2　对话板的操作过程

在分析对话板的工作过程前，先熟悉对话板的操作过程将有助于程序的分析和理解。

对话板的操作过程如下：

1、开始

在等待输入姓名过程中，OPERATOR 栏出现闪烁的光标。假如我们输入姓名"LI MING"，按回车键后，则清两栏、显示

"OK!

LI MING

IS YOU"

延时后清栏、显示

"NOW WE START!"

延时后清栏、显示

"F BASIC

NEED?"

清四个提示栏，栏中依次显示"OK、NO、HELLO、END"四项提示。等待操作者选项。

2、选择询问中键入"NO"的操作过程

在选择询问中若按 F2 键则在 OPERATOR 栏输入了字符"NO"，回车后，清栏、显示

"CALCULATOR

NEED?"

按 F2 键、回车后、清栏、显示

"MUSICBOARD

NEED?"

按 F2 键、回车后、清栏、显示

"MESSBOAD

NEED?"

按 F2 键、回车后、清栏、显示

"F BASIC
NEED?"

又返回第一个询问,以后重复以上过程。

3、选择询问中键入"HELLO"的操作过程

在任一项询问中按 F3 键,则清栏、显示

"YES"

延时后清栏、显示

"HELLO!
HOW ARE YOU"

延时后清栏、显示

"I'M CNEDC FAMILY
COMPUTER SYSTEM"

延时后清栏、显示

"CNEDC SHENZHEN CO.
IS GLAD TO SERVE
FOR YOU.
THANKS!"

延时后清栏、显示

"CNEDC SHENZHEN CO.
TEL:361105 365402
CABLE:4703"

延时后清栏、显示

"HAVE A REST?"

等待操作者输入"OK"或"NO"。

若按 F2 或键入"NO",回车后清栏、显示

"OK!
NOW WILL BE
CONTINUE!"

延时后清栏、显示

"NOW WE START!"

延时后清栏、显示

"F BASIC
NEED?"

重复选项询问操作。

若按 F1 或键入"OK",回车后清栏、显示

"OK!
PRESSING RESET
COMPUTER WILL WORK
AGAIN"

电脑进入死机状态,只有按复位键进入复位画面才能重新工作。

4、选择询问中键入“END”的操作过程

在询问中按 F4 键，则 OPERATOR 栏显示“END”，回车后清栏、显示

“OK
NOW WE...”

延时后清栏、显示

“HAVE A REST.
OR”

延时后清栏、显示

“DO YOUR HOMEWORK”

延时后清栏、显示

“BYEBYE”

延时后返回复位画面。

3·3·3 对话板的工作过程

一、操作者姓名的输入过程

等待操作者输入姓名的处理程序入口为＄D0E8。按功能可把这段程序分为两大部分：第一部分为等待输入处理程序＄D0E8～＄D0FA；第二部分为等待时间结束处理程序＄D0FB～＄D101。

1、＄D05D 子程序简介

等待输入姓名过程的主要操作是调用＄D05D 子程序。

＄D05D～＄D06D 是对话中操作者输入内容的判断、检索、处理专用子程序。它的主要任务是：

(1) 调＄D4BA 子程序判断功能键使用情况，若按 F1～F4 功能键则进行功能键定义的检索、翻译处理，检索出功能键定义的各字符 ASCII 码存入＄304 和＄63D 开始的各单元中；若不是功能键则直接返回。

(2) 调用＄CF78 子程序

检查是否按回车键，若是回车键则转＄CFA5；

调＄EF2C 子程序，对键码进行判断处理；

调＄F546 子程序，控制 OPERATOR 栏的光标闪烁；

调＄D77B 子程序，改变 OPERATOR 栏四个方块的状态，令其闪烁；

调＄D709 子程序，调整 SP 块的状态、坐标及 PPU 配色，令信号灯闪烁。

2、操作者姓名的输入

(1) 当输入姓名的第一个字母时，按“L”键则通过调＄8003 子程序读出键码，有＄37＝H4C；

(2) 转＄D0E8 调＄D05D、再调键码判断处理子程序＄EF2C。因＄37＝H4C 故跳转＄F03E 输入字符的处理程序，把键码送入＄63D+＄2F 所指的单元(＄2F 为 OPERATOR 栏送显字符计数器)。然后令＄FF＝4 发出 OPERATOR 栏送显字符时的“铛”声。再调＄F288 把送显 OPERATOR 栏的字符数据送入三页显示。最后调整 OPERATOR 栏光标的位置，令字符计数器＄2F 加一。

(3) 当输入“LI MING”后，六页有关单元的内容依次如下：

$63D=4C “L”

$63E=49 “I”

$63F=20 空格

$640=4D “M”

$641=49 “I”

$642=4E “N”

$643=47 “G”

$2F=7

3、按回车键后的处理

(1) 输入姓名后，按回车键，$37=H0D，由$CF83控制转$CFA5进行按回车键的处理。首先令$FE=H10以发出按回车键后的“嘟噜”声；令$FC=0以关闭轻微的“哆哆”声；然后令$644=H0D、$2F=8、$60=0，为重新设置程序入口做好准备。

(2) 令$60=H0D、查表3-4，转$CF1F，把$63D～$64F的内容送入$500～$512中，保存操作者的姓名。然后清两栏、送显“OK！”，令$60=0。

(3) 令$60=6、$C3=H2F、查表3-4、转$D2A4，清两栏后；从数据区$C97C开始取数送显。首先送显“OK！”、换行后，读到A=$C980=C0时转$D37E，设置数据区首址于$C0、$C1中，令这两个单元指向操作者姓名的首址$500，并令$C2=1置送显名词标志。因此，在以后的处理中，由$D2D7转名词送显程序$D3B0，把操作者的姓名送入COMPUTER栏第二行显示。当读到$507时，A=H0D、则令$C2=0、换行，在下一次中断时又转$D2DA叙述字符送显程序，送显“IS YOU.”，从而显示出

“OK!

LI MING

IS YOU.”

的画面。

(4) 显示延时，令COMPUTER栏的四个方块闪烁。延时结束后，清两栏，送显

“NOW WE START!”

令$FE=2发出送显完毕的“叽哩咕噜”声。

(5) 令$60=H17、查表3-4，设置程序转移入口为$D0DA，进入选择、询问对话阶段。操作者姓名的输入阶段结束。

二、询问选项工作过程

1、执行$D0DA程序，令$88=H0F、$43=0、设置询问选项进程标志。$43与询问内容的对应关系见表3-17。

表3-17 $43取值与询问内容的对应关系

$43单元取值	COMPUTER栏询问的内容
0	F BASIC
1	CALCULATORBOARD
2	MUSICBOARD
3	MESSAGEBOARD

2、令$60=6，$C3=8、查表3-4转$D2A4，显示延时结束后清两栏，由$C3=8、查表3-16得数据区首址为$C607。送显双引号后，第二个数据为HC2，因而送显“F BASIC”；然后

再送显一个双引号，换行后送显“NEED?”；最后令＄FE＝2、＄2A＝H58、＄2D＝3，等待输入延时，延时结束后令＄60＝0。

3、令＄60＝H0C、＄6B＝03、查表3－4设置程序转移入口为＄D44A。由＄6B＝03控制令四个提示栏显示提示内容，送显提示名词的顺序依次是：END、HELLO、NO、OK。

在提示栏的操作中，由＄6B控制对提示栏的操作种类，它们的对应关系见表3－18。

表3－18　＄6B取值与提示栏操作的对应关系

＄6B单元的取值	对提示栏操作的种类
0	清四个提示栏
1	只显“OK、NO”
3	显示“OK、NO、HELLO、END”

4、令＄60＝H15、查表3－4得程序转移入口为＄D06E。执行＄D06E程序，首先令＄2D＝HFF，置计数延时，令＄FC＝H68发出等待输入的“哆哆”声；然后调＄D05D按键判断检索处理子程序检查按键情况；最后转＄D0F7检查延时器＄2D单元，若＄2D＝0则返回复位画面。

5、当我们按F2键时，＄37＝HC2，经调＄D4BA功能键翻译子程序，把F2代表的“NO”字符送入OPERATOR栏显示，并同时送入＄63D、＄63E，有＄63D＝H4E、＄63E＝H4F，令＄FE＝H40发出“发拉”的声响，令＄FC＝H68发出等待输入的“哆哆”声，等待按回车键。

6、按回车键后、＄37＝H0D，则令＄FE＝H10、发出“嘟噜”声响，令＄FC＝0关“哆哆”声；再调＄D05D、令＄63F＝H0D、＄60＝0；然后把＄63D～＄63F的内容送入＄550～＄552单元保存；最后令＄43加一，置询问进程标志，并令＄88＝H10。

7、令＄60＝6、＄C3＝8、转＄D2A4清两栏；然后由＄C3＝8，查表3－16得数据入口为＄C607，经＄D2C4子程序处理，送显

“CALCAULATOR
NEED?”

以后的操作过程同“F BASIC NEED?”询问相同（分析从略）。

关于选择“HELLO”、“END”的处理过程与“NO”基本相同，留给读者分析。

3·3·4　重要子程序介绍

为便于分析对话工作过程，特对几个重要子程序简介如下：

一、操作者输入内容检验程序＄CFC4

＄CFC4～＄CFDB是操作者输入内容检验子程序。该程序有两个作用：第一检验操作者输入的内容——系统名词是否合法；第二检索出系统名词的代号供＄D07C程序判断以转入对应选项的处理程序。系统名词的数据首址和代号的对应关系列于表3－19。

表3－19　系统名词的代号和数据区首址对照表

系　统　名　词	数据区首址	＄00单元中的名词代号	对应的＄62单元的内容
END	CAB7	03	0B
OK	CABB	04	00
NO	CABE	05	00
F BASIC	CAC2	06	00
CALCULATOR	CACD	07	00
MUSICBOARD	CAD9	08	00
MESSBOARD	CAE4	09	00

系　统　名　词	数据区首址	$00 单元中的名词代号	对应的 $62 单元的内容
END	CAEE	0A	0B
HELLO	CAF2	0B	2C
HI!	CAF8	0C	2C
HELLO	CAFD	0D	2C
HAHA!	CB03	0E	2C
0D	CB09	0F	00
0D·	CB09	10	00
BASIC	CB0A	11	00
CAL·	CB10	12	00
MUS·	CB15	13	00
MES·	CAA1	14	00

现在我们对照表 3－19 分析 $CFC4 子程序。假如我们欲选择“HELLO”功能，则按 F3 键，有

$63D＝48　　　　$63E＝45

$63F＝4C　　　　$640＝4C

$641＝4F　　　　$642＝0D

$CFC4 子程序的检验是从各板简写开始的。它首先令 $00＝H14、X＝ $00，调检索子程序 $CFE6，进行以下处理：

1、在 $CFE6 中，根据 X 的取值，从数据区 $CA8D～ $CAB6 查出名词代号为 H14 的数据区首址为 $CAA1(见表 3－19)，把它存入 $02、$03 中；然后令 X＝0、取 $63D＋X 的内容与 $CAA1＋X 的内容相比较，若不等则令 A＝FF，返回调用程序 $CFCA。

2、令 $00 减一，再令 X＝ $00、再调 $CFE6 子程序进行第二次检验。

3、直到 $00＝H0D 时，查出 X＝H0D 的数据区首址为 $CAFD；令 X＝0、如有 $63D＋X ＝ $CAFD＋X 则令 X＝X＋1，并把比较的数据送入 $550＋X；若第二个数据也相等再比较第三个，直到各位数据都相同，且读到 A＝H0D 则表示比较结束，操作者输入的内容合法。令 A ＝0 返回调用程序 $CFCA。

4、由于 A＝0 故由 $CFCF 转 $CFDC，以 $00 单元的内容为变址，由 $CD01～ $CD15 数据区查出对应的数据(见表 3－19)赋给 $62 单元，以改变 $60 单元的赋值数据区，转向操作者选定内容的处理程序。

二、回车键处理子程序 $CFA5

当操作者输入完毕、按回车键后则必须在 $63D 以后的单元中查出本次输入内容的末尾，在末尾的后一个单元置入 H0D，以表示输入结束，以便在 $CFC4 输入内容检验子程序中作为结束检验的标志。因此，要通过 $CFA5～ $CFC3 子程序处理以置入 H0D，该子程序较简单，留给读者分析。

三、按键判别处理子程序 $EF85

$EF85～ $F099 是一个按键综合判断处理专用程序。它分别对方向键、回车键、功能键、Del 键、Ins 键进行判别处理。最后，以空格键码 H20 划线，对键码大于 H20 的字符键和小于 H20 的操作键进行分别判断处理。各类按键的处理程序划分如下：

- $EF97～ $EF9F 是按右移键的处理程序；
- $EFA4～ $EFAC 是按左移键的处理程序；

· \$EFB1～\$EFC2 是按上移键的处理程序；
· \$EFC7～\$EFD8 是按下移键的处理程序；
· \$EFDD～\$EFF7 是按回车键的处理程序；
· \$F007～\$F019 是按 Ddl 键的处理程序；
· \$F01E～\$F030 是按 Ins 键的处理程序；
· \$F03E～\$F099 是按空格键和字符键的处理程序。

\$EF85 是一个通用的按键综合判断处理程序，各板字符的输入和大部分屏幕编辑处理都在该程序中完成。因此请读者务必把以上各段程序看懂弄通。

四、功能键判断处理子程序 \$D4BA

\$D4BA～\$D50B 是功能键判断处理子程序。当按功能键时，由该子程序翻译出所按功能键代表的内容送入三页显示，其工作过程如下：

1、由 \$D4C5～\$D4CB 设置三页功能键定义框架。其框架结构为：\$301、\$302 为送显 PPU 的 OPERATOR 栏的首地址；\$303 为送显字符的数量，其值为 H05；\$304～\$308 全部置为 H20。

2、由 \$D4CC～\$D4D3 程序段通过调用 \$D48F 查出功能键定义字符的 ASCII 码送入 \$304 和 \$63D 开始的各单元。

3、由 \$D4D4～\$D4E8 把 \$63D～\$7D3 置为 H20，把 \$303～\$322 中的空余单元全部置为 H20。

4、由 \$D4E9～\$D4F7 程序把 \$304 以后单元中所存的功能键定义码送入 \$63D 以后的单元，并调整 OPERATOR 栏送显字符的 PPU 地址和光标的坐标。

5、由 \$D4F8～\$D50A 程序令 \$300=H15、\$303=H12、\$316=00、\$FE=H40，即向三页送显功能键的定义内容并发出按键声。

3·3·5 对话过程中的音响

对话中由七种不同的效果音响标志着不同的操作完成情况。现将其归纳于表 3－20。

表 3－20 对话工作过程中的音响数据及应用表

发声控制单元和数据	声 响 特 点	使 用 场 合
FF=02	滴滴声	COMPUTER 栏送显字符时
FF=04	铛铛声	OPERATOR 栏送显字符时
FE=02	叽哩咕噜声	COMPUTER 栏每次送显完毕
FE=04	《七宝奇谋》效果音	从复位画面进入对话板时
FE=10	嘟噜声	按回车键时
FE=40	发拉声	按功能键时
FC=68	轻微的"哆哆"声	等待操作者输入时

音响的发声过程留待以后讨论。

3·3·6 对话板工作小结

对话板的工作过程在四种操作中反复进行：

1、执行 \$D2A4 转 \$D2C1 清两栏。

2、执行 \$D2A4 转 \$D2C4 进行 COMPUTER 栏的操作。

3、执行 \$D44A 对提示栏进行送显或清除操作。

4、执行 \$D06E 对操作者的输入进行检查、判断处理。操作者每次输入完毕令 OPERA-

TOR 栏的四个白色方块闪烁。

另外，作用于信号灯的 SP 块仍由中断计数器控制，随机地改变状态和坐标。

每次清两栏前要对 PPU 的配色单元 $ 3F0C～$ 3F0F 置入 H0F，每次送显前要对 $ 3F0D 置入 H30，以显示黑底白色的字符。

通过对话板的工作过程分析，我们可以从中学习、掌握查表法送显字符的编程方法和部分清屏技术。

3·4 计算板的工作过程

在对话板工作中，当询问

"CALCULATOR NEED?"

时按 F1 键或其它询问时键入"CAL·"回车后，都可进入计算板。

在计算板中，可进行"＋"、"－"、"×"、"÷"及四则混合运算，运算精度可达八位有效数字。数据运算范围为±(0.00000001～99999999)。当计算结果中的有效数字多于八位时，若整数部分不多于八位系统会自动删除八位以后的小数部分；当整数部分超过八位时会显示"OVERFLOW"(溢出)信息；当输入算式中有错误时会显示"WRONG"(错误)信息。

整板有八行空间供输入算式。算式之间可用空格间隔而连续输入。

在计算板，F1～F8 八个功能键的定义内容依次为

"＋"、"－"、"×"、"÷"、"("、")"、"."、"＝"

F8 键(等号键)为运算命令键，只要按 F8 键，则把羽毛笔前面的算式进行数学运算后输出结果。

按回车键后可换一行书写；按"Home"键可把羽毛笔移到上一行的最左端；按"Shift＋Home"键可以清屏；按方向键可上下左右移动羽毛笔；按"Ins"键可在羽毛笔处插入空格，插入空格的数量以本行为限；按"Del"键可删除羽毛笔前的一个字符，同时羽毛笔后的内容全部前移一格，前移的字符也以本行为限；按"Esc"键表示 CAL·板工作结束，返回对话板。

另外，在计算板工作中，还始终有以进行速度的打击音乐伴奏，从而更增添了 CAL·板的魅力。

计算板的工作过程共分五大阶段，分别为：

· 计算工作状态的建立
· 算式输入
· 算式检验
· 算式运算
· 运算结果输出

下面我们分节讨论。

3·4·1 计算板工作状态的建立

在对话板工作中，当询问"CALCULATOR NEED?"时按 F1 键、回车后则进行以下操作：

1、清两栏，COMPUTER 栏送显"OK!"、延时、清两栏、送显

"CALCULATOR"

START.

2、令＄60＝9、查表 3－4、转＄CF06、关显示、清屏、清二页。

3、令＄60＝H12、查表 3－4 转＄D107、令＄62＝＄87＝H0D、设置＄60 单元的赋值地址为＄CC0B。

4、令＄60＝H1C、查表 3－4 转＄D10F、令＄63D～＄7D3＝0。

5、令 X＝8、调＄F4B3 定义八个功能键，把 F1～F8 定义为：

F1＝H2B	F2＝H2D	F3＝HB5	F4＝HB6
“＋”	“－”	“×”	“÷”
F5＝H28	F6＝H29	F7＝H2E	F8＝H3D
“(”	“)”	“.”	“＝”

6、调＄D1FE 把 CAL·板数据库＄7100～＄71D7 中的数据送入 CAL·板工作区＄63D～＄714。（当第一次使用 CAL·板时，数据库中全部为 H20）

7、令＄FC＝1，启动背景音乐开始伴奏。

8、令＄A0～＄DF＝0，为数据运算做好准备。

9、令＄FE＝1 发出换板时的效果音响。

10、转＄F50F 把定义向上羽毛笔的八个 SP 块的数据写入二页。开始位置的羽毛笔数据如下：

＄2C0＝3C	＄2C4＝3C
＄2C1＝E0	＄2C5＝E1
＄2C2＝00	＄2C6＝00
＄2C3＝22	＄2C7＝2A
＄2C8＝44	＄2CC＝44
＄2C9＝E2	＄2CD＝E3
＄2CA＝00	＄2CE＝00
＄2CB＝22	＄2CF＝2A
＄2D0＝4C	＄2D4＝4C
＄2D1＝E4	＄2D5＝E5
＄2D2＝00	＄2D6＝00
＄2D3＝22	＄2D7＝2A
＄2D8＝54	＄2DC＝54
＄2D9＝E6	＄2DD＝E7
＄2DA＝00	＄2DE＝00
＄2DB＝22	＄2DF＝2A

11、令＄60＝H0E、＄6B＝2、查表 3－4 转＄CEF8、令＄8A＝2、＄60＝0。

12、执行中断程序中的＄CDD3，由＄8A＝2 计算出绘图数据区首址为＄D9A5，把它存入＄00、＄01 单元。调＄EE43 显示 CAL·的板名、功能键定义提示、向 PPU 配色单元置入配色数据。

13、令＄8A＝5、把绘图数据区首址＄DA22 存入＄00、＄01 单元。调＄EE43 绘制 CAL·板的边框、横线等。至此，CAL·板画面绘制完毕。CAL·板的工作画面数据见图 3－5，背景 0 页配色分布见图 3－6，CAL·板的 PPU 配色分布见表 3－21。

14、令＄60＝2、查表 3－4 转＄DC77，进入 CAL·板的工作状态。

列次	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	行次
2120	/		/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	1
2140	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	2
2160	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	3
2180	/	/	F3	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F7	/	/	4
21A0	/	/	F4					EF	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	F0					F4	/	/	5
21C0	/	/	F4					EE	C	A	L	C	U	L	A	T	O	R	␣	B	O	A	R	D	EE					F4	/	/	6
21E0	/	/	F4					F1	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	F2					F4	/	/	7
2100	/	/	F4																											F4	/	/	8
2120	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	9
2140	/	/	F4																											F4	/	/	10
2160	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	11
2180	/	/	F4																											F4	/	/	12
21A0	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	13
21C0	/	/	F4																											F4	/	/	14
21E0	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	15
2200	/	/	F4																											F4	/	/	16
2220	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	17
2240	/	/	F4																											F4	/	/	18
2260	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	19
2280	/	/	F4																											F4	/	/	20
22A0	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	21
22C0	/	/	F4																											F4	/	/	22
22E0	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	23
2300	/	/	F4																											F4	/	/	24
2320	/	/	F4		F	1	:	+	·		F	2	:	—			F	3	:	×			F	4	:	÷	F9	FB		F4	/	/	25
2320	/	/	F4		F	5	:	(			F	6	:	)			F	7	:	·			F	8		=	FA	FC		F4	/	/	26
2340	/	/	F5	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F8	/	/	27
2360	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	28
2380	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	29
23A0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	30
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	

图 3-5 CAL·板工作画面

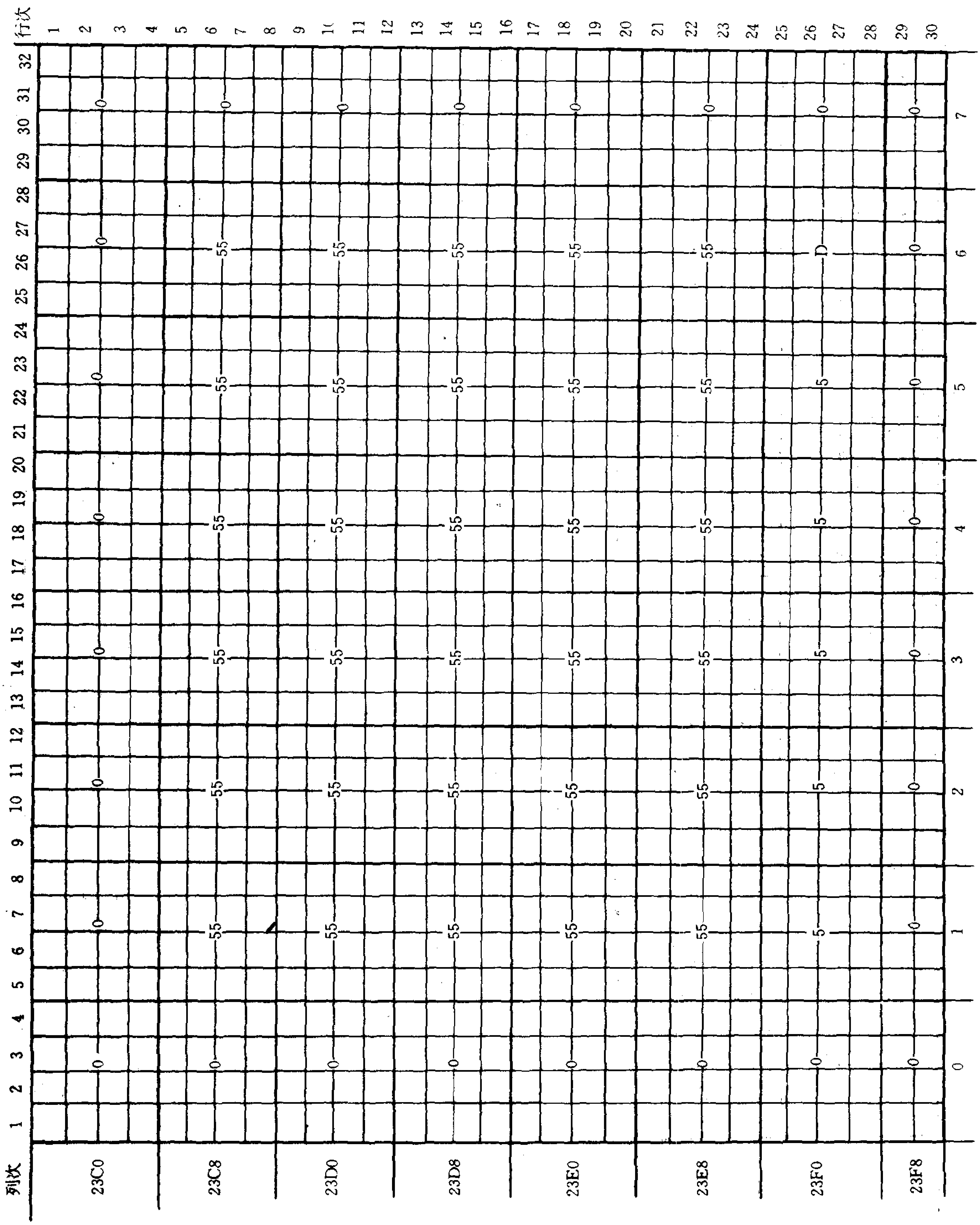


图 3—6 CAL·板背景 0 页配色分布图

表 3－21　CAL·板 PPU 配色分布一览表

PPU 地址	颜色代码	PPU 地址	颜色代码
3F00	0F	3F0C	0F
3F01	30	3F0D	30
3F02	2C	3F0E	2C
3F03	02	3F0F	02
3F04	0F	3F10	0F
3F05	30	3F11	26
3F06	36	3F12	30
3F07	26	3F13	15
3F08	0F	3F14	0F
3F09	30	3F15	29
3F0A	36	3F16	30
3F0B	07	3F17	1A

3·4·2　计算板的算式输入过程

我们以典型的算式

(3982＋56789876)÷325×1.2－6

为例，介绍各阶段的操作过程。

通过上一节的讨论，我们已经了解 CAL·板在中断程序中的工作程序入口为＄DC77。现在我们从 DC77 开始讨论算式的输入过程。

一、保存键码

算式的第一个字符是“(”，按 F5 键，通过调＄8003 读键程序读出键码存入＄37 单元，有＄37＝H28。

1、执行＄DC77，因现在＄79＝＄7D＝＄D8＝＄41＝0，所以转＄DCB0，由＄DCB2 调＄EF2C，因＄23＝0 转＄EF5B，把键码存入键码暂存器＄38，然后转＄EF85 按键综合判断处理子程序。

2、在＄EF85 子程序中，首先把键码存入按键工作单元＄39、＄40；然后转＄F051 把键码存入＄63D＋＄2F 单元，输入第一个算符时＄63D＝H28；最后令＄FE＝H40、发出“发啦”按键声，令＄23＝H10、返回＄DCB2。

3、返回＄DCB2 后，检查＄B0、＄3A 单元，现在＄B0＝＄3A＝0。本次中断处理结束。

二、羽毛笔画圈

羽毛笔画圈操作在接受字符的下一次中断开始，画圈过程由＄23 单元控制，经过 15 个中断周期完成。下面我们讨论画圈经过。

1、在第二次中断时(指输入字符后的第二次中断，下同)、＄23＝H0F，由＄DC77 转＄DCB0，由 DCB2 调＄EF2C，现在因＄23＝H0F 故转＄EF50，由 EF50 调＄F4E1。

2、＄F4E1～＄F545 是羽毛笔定义、移位和画圈专用子程序，在 CAL·、MUS·、MES·三板的字符输入过程中都要调用它。画圈的控制方法是：从＄2C0、＄2C3 中取出羽毛笔当前的 X、Y 坐标值、再从＄EEA9～＄EEC9 羽毛笔坐标增量数据区取出坐标增量值、与羽毛笔当前坐标值相加后存入＄00、＄01 单元；然后转＄ECBE 定义 SP 块子程序，令羽毛笔在新的坐标上显示。

3、在羽毛笔的画圈过程中，其 X、Y 坐标的变化过程如表 3—22，羽毛笔的位移轨迹见图 3—7(以上仅以画第一个圈为例)。

表 3—22　羽毛笔画圈时的 X、Y 坐标变化表

$23	$2C3	$2C0	$23	$2C3	$2C0
10	22	3C	08	23	39
0F	21	3E	07	24	3A
0E	1F	3D	06	25	3B
0D	1E	3C	05	26	3C
0C	1F	3B	04	25	3D
0B	20	3A	03	24	3E
0A	21	39	02	23	3D
09	22	38			

三、送显字符

1. 当经过 14 次中断后 $23＝1(每中断一次羽毛笔画一点、$23 减一)，羽毛笔画圈完毕。由 $DCB2 调 $EF2C、因 $23 = 1 转 EF54、再转 $F076 进入字符送显处理阶段。

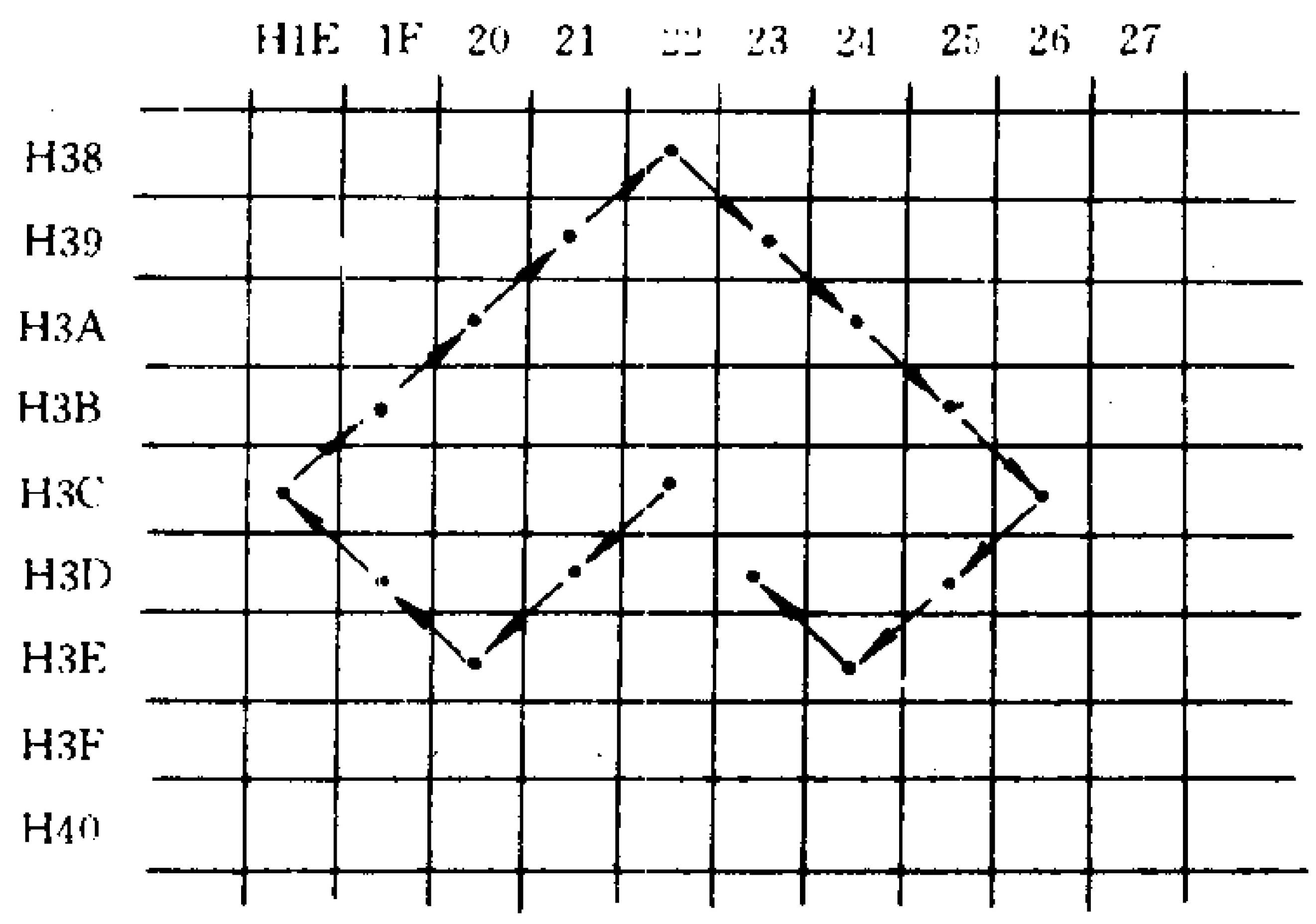


图 3—7　羽毛笔画圈轨迹示意图

2、首先调 $F288 设置字符送显框架。框架结构为

$301＝20　　　$302＝E4

$303＝01　　　$304＝20

$305＝00

然后把 $38 单元暂存的键码 H28 送入 $304。

3、令 $3A 加一，置本次输入字符有效。

4、调 $F210 子程序调整 PPU 字符送显地址，令 $34＋1＝HE5。

5、调 $F115 子程序调整羽毛笔的显示坐标，令 $30＝H2A、并令输入字符计数器 $2F 加一。

6、调 $F50F 子程序令羽毛笔在新的位置上显示。返回 $DCB2。

7、因 $3A＝1、调 $F473 向 $6ED＋ $2F 单元置入 H20。至此，一个字符的输入操作完毕。

整个算式的输入过程完全相同，不多赘述。算式输入结束后，有关单元内容如下：

$63D＝28　　　$63E＝33

$63F＝39　　　$640＝38

$641＝32　　　$642＝2B

$643＝35　　　$644＝36

$645＝37　　　$646＝38

$647=39	$648=38
$649=37	$64A=36
$64B=29	$64C=B6
$64D=33	$64E=32
$64F=35	$650=B5
$651=31	$652=2E
$653=32	$654=2D
$655=36	$656=20
$2F=19	$30=2A
$31=4C	$32=21
$34=25	

四、算式输入过程小结

1、按字符键后，首先把字符的 ASCII 码送入 $63D+ $2F，然后令 $23=H10 控制羽毛笔画圈。

2、令 $FE=H40、发出“发啦”声响。

3、画圈完毕把字符送入三页显示，令字符计数器 $2F 加一。

4、令羽毛笔后移一格，并把 $63D+ $2F 置为 H20。

5、重要子程序

·按键检查判断处理子程序 $EF2C；

·羽毛笔画圈控制子程序 $F4E1；

·设置三页送显字符框架子程序 $F288。

3·4·3 算式检验工作过程

算式检验工作从输入等号开始。算式检验的任务是：检查输入的算式是否合法，若有错误则打印“WRONG”出错提示；若合法则置算式合法标志，令 $41=1、结束检验过程。

检验工作过程如下：

一、等号的送显过程

当算式输入完毕，按 F8 键时，由读键程序读出键码送入 $37 单元，令 $37=H3D。随后的羽毛笔画圈和送显“=”的过程与输入字符基本相同，差别是：送显完成后，因 $38=H3D=$EE68，因而转 $F09A、令 $B0=1，置运算指令发出标志。同时，把 $63D 以后单元中算式的最末单元的下一单元置入 H20，即令 $656=H20。返回调用程序 DCB2。

二、算式检验过程

因 $B0=1 则由 $DCB9 调算式检验程序 $DE5A。

$DE5A～ $DF8E 是 CAL·板算式检验专用程序，它所控制的操作如下：

1、调 $DF8F 子程序把 CAL·板算式暂存库 $7600～ $76FF 清零，令其等于 H20。

2、调 $E3D2 子程序把 $418～ $422 单元置为 H20。

3、由 $DE60～ $DE78 处理，找出本次输入算式的第一个字符位置。

4、由 $DE7C～ $DE8B 处理，把 CAL·板的当前算式送入暂存库 $7600～ $76FF。

5、转 $DE8F 算式检查子程序，分步检查算式是否合法。

$DE8F～ $DF8E 是算式检查判断专用执行子程序。在该程序中，由 $B2～ $B5、$BA、

$BC～$BE 八个单元作为各算符合法标志。算式中各种元素的判断处理程序和合法标志见表 3－23。

表 3－23　CAL・板算式检验工作表

算式元素	处 理 程 序	算 式 合 法 标 志
(	DF2E～DF44	$BC=1、$BD=0、$BE=0
)	DF49～DF69	$BC=0、$BD=0、$BA=1
+、-、×、÷、	DF00～DF29	$B4=1、$B5=0、$BA=0 $BD=1、$BE=0
数据	DECF～DEEC	$B3=0、$BD=0、$BE≠0
小数点	DEB3～DEC6	$B3=1、$B5=1
等号	DF6E～DFCD	$B4=0、$BC=0、$BD=0

算式的合法性检查程序虽然较长，但属于顺序结构，较易理解，故分析从略。

当算式检查合法后，令 $41=1，发出运算指令。

另外，算式不合法（有错）的处理程序在 $DF89～$DF8E，出错处理子程序在 $DE01～$DE30，它们的作用是把出错信息送入 $3F0 开始的各单元中，以备送显。

算式检验结束后，算式暂存库的内容如下：

$7600=28　　$7601=33
$7602=39　　$7603=38
$7604=32　　$7605=2B
$7606=35　　$7607=36
$7608=37　　$7609=38
$760A=39　　$760B=38
$760C=37　　$760D=36
$760E=29　　$760F=B6
$7610=33　　$7611=32
$7612=35　　$7613=B5
$7614=31　　$7615=2E
$7616=32　　$7617=2D
$7618=36　　$7619=20

3・4・4　算式的运算过程分析

CAL・板的算式运算处理在复位主程序中完成。当算式检验合法后，由 $41=1 控制调用 $DFA6 运算程序，开始算术运算处理。

一、CAL・板的运算分类与运算顺序

在算术运算中，CAL・板把运算分为三度优先级，从最高级开始依次为：

・括号运算（第一类或第一级）

・乘、除运算（第二类或第二级）

・加、减运算（第三类或第三级）

运算开始，首先检查算式中是否有括号运算。当检索到括号时则把括号内的算式按乘、除和加、减分为两类，先进行乘、除运算后进行加减运算。

当第一个括号中的内容运算完毕，保存结果后再检查算式中是否还有括号运算，若有则进

行第二个括号的运算，直到算式中所有括号都运算完毕为止。

当算式中无括号或括号全部运算完毕后则进行第二级运算，按"×÷"算符依序进行，直到同级算符运算完毕为止。

当算式中无第一、第二级算符或这两级算符运算完毕时则进行第三级算符运算。当加、减运算全部完成后则整个算式运算结束。

二、运算处理中使用的主要内存单元介绍

为便于分析，特对 CAL·板算式运算中使用的主要内存 RAM 单元及作用介绍如下：

1、$400～$40A：第一运算数据存放单元。用于存放加数、被减数、被乘数、被除数等。

2、$410～$41A：第二运算数据存放单元。用于存放另一个加数、减数、乘数、除数等。

3、$420～$42A：中间运算结果暂存器。

4、$430～$43A：当前运算结果暂存器。

5、$3F0～$3FA：最终运算结果暂存器。

在 $400～$43A 中的 $409、$419、$429、$439 单元用于存放当前数据的小数点后的小数的位数；$40A、$41A、$42A、$43A 单元用于存放当前数据的符号。

6、$38C～$3EF：用于存放各括号中的运算结果。每个括号占用连续的 10 个单元，最多可以存放 10 个括号的运算结果。括号的序号与占用的内存单元对照情况列于表 3－24。

表 3－24 括号序号与占用内存单元对照表

括号序号	变址指针 $C5	括号运算结果的存放地址	括号序号	变址指针 $C5	括号运算结果的存放地址
1	80	38C～395	6	B2	3BE～3C7
2	8A	396～39F	7	BC	3C8～3D1
3	94	3A0～3A9	8	C6	3D2～3DB
4	9E	3AA～3B3	9	D0	3DC～3E5
5	A8	3B4～3BD	10	DA	3E6～3EF

7、$8E～$96：用作各数据的 ASCII 码还原为数据的暂存器。

8、$97～$9F：当前运算结果暂存器。

三、运算处理过程

运算处理共分五个阶段：

- 读入数据阶段
- 运算前予处理阶段
- 运算处理阶段
- 运算结果处理阶段
- 转入下一级运算阶段

1、读入数据阶段

读入数据阶段的主要任务是：把两个运算数分别存入相应的工作单元，并进行数据还原。工作过程是：首先把参与运算的两数据分别装入 $400～$408、$410～$418 单元，并令两数的个位（小数则是最末位）分别存入 $408、$418 中，其它各位依次向前存放。当数据是小数时则去掉小数点，把小数的位数存入 $409、或 $419 单元；其次把两数的符号分别存入 $40A、$41A，若是正数则置入 H20，若是负数则置入 H2D；最后把 $400～$408、$410～$418 各单元的数据由 ASCII 码的表达形式还原成一位数字仍存于原单元（还原的方法是令各单元的内容与 H0F 相与或减 H30）。另外，当从 $76×× 单元取出数据后还要及时将该单元置为

H20。

2、运算前予处理阶段

予处理阶段的主要任务是：根据运算种类把两运算数据调整成运算处理程序要求的形式。处理过程如下：

(1) 根据运算种类决定运算结果的符号存入＄43A 单元。

(2) 若进行小数的加、减运算，则要对数据进行移位处理，使两运算数的个位对齐。

(3) 进行乘、除运算时，要检查两数是否有一个为零。乘运算中若有一数为零则令结果为零，跳过运算处理程序。除法运算中，若除数为零则给出错误信息、结束运算；若被除数为零则同乘法处理相同。

(4) 当进行减运算时，要比较两数的大小，使绝对值大的数装入＄400～＄408，同时决定运算结果的符号存入＄43A。

3、运算处理阶段

经予处理使两运算数的形式达到运算程序的要求后，转入相应的运算程序进行运算，并把运算结果转换为 ASCII 码存入＄430～＄438 中。

4、结果处理阶段

一次运算结束后，首先判断结果是否溢出；然后根据＄409、＄419 的内容决定运算结果的小数位数存入＄439，并查出小数点的位置把小数点(H2E)置入＄430～＄438 中的某一单元。

5、转入下一类运算阶段

当属于第一、二类运算时，当前运算完成后，还要检查算式中是否还有同类运算。若有则继续进行本类运算；若没有则将本类运算结果送入中间结果暂存器＄420～＄42A，转入下一类运算。

当是第三类运算时，本类运算完成后则转运算结束处理。

四、运算处理程序简介

由于运算处理程序较长，大约占用 3.2K 内存，分布在＄DC63～＄E3DC 和＄E6E8～＄EC59 两段 ROM 空间，因而阅读理解较困难。为此特将运算程序按功能划分为若干程序段、注明其主要作用，以供分析程序时参考。

运算程序分段简介如下：

1、运算处理主控程序＄DFA6～＄DFCB

其中：＄DFA6～＄DFB7 为有关工作单元初始化处理。

＄DFB8～＄DFC4 为括号运算程序。

＄DFC5～＄DFC7 为乘、除运算程序。

＄DFC8～＄DFCA 为加、减运算程序。

2、运算类别转移控制程序＄DFD3～＄DFE7

3、括号运算处理程序＄DFEC～＄E065

4、加法运算子程序＄E6E9～＄E74A

5、减法运算子程序＄E74B～＄E7A2

6、运算前予处理子程序＄E7A3～＄E7B7

7、乘法运算子程序＄E7B8～＄E885

8、除法运算子程序＄E886～＄E9CB

9、运算结果溢出处理子程序＄E9CC～＄E9F0

10、运算数由 ASCII 码还原为单位数的处理子程序＄E9F1～＄EA51

11、小数的运算前处理程序＄EA52～＄EA8A

12、小数结果整形及加入小数点子程序＄EA8B～＄EADE

13、判断运算数和运算结果是否为零子程序＄EADF～＄EB0F

14、控制装数单元前后移位的系列子程序＄EB34～＄EBC5

15、两运算数互相交换内容子程序＄EBC6～＄EBE5

16、运算结果的符号处理子程序＄EBFD～＄EC4B

17、运算符增量检索子程序＄E1F8～＄E228

18、运算符减量检索子程序＄E229～＄E253

19、第二、第三类运算符分类检索子程序＄E254～＄E286

20、括号运算子程序＄E287～＄E38A

五、括号运算过程

运算开始后，令＄00＝H28、调＄E1F8 运算符增量检索子程序，检索左括号“(”，当检索到后则调＄DFEC 子程序进入括号运算处理。

1、令＄00＝H29、调＄E1FC 检索右括号“)”，查出括号运算式长度(括号内算式字符数)存入＄C1 单元中。

2、把＄76××中左括号存储单元置入 H41，右括号存储单元置入 H42，对本例有＄7600＝H41、＄760E＝H42。

3、调＄DFD3，首先检索括号中有无第二类运算符。若没有则调用＄E066 第三类算符处理子程序进行加、减运算。

4、当第一个算符运算完毕后，再检查括号中是否还有第二、第三个算符，若有再进行以后算符的运算……，直到所有算符运算完毕则结束当前括号运算。

5、把存于＄430～＄43A 中的运算结果送入括号结果暂存单元＄38C～＄395 中，并把括号运算结果暂存地址指针＄C5 的内容存入左括号的下一个单元中，令＄C5 加 H0A，使其指向第二个括号运算结果暂存单元首址。尔后再检查后面的算式中是否还有括号。若还有括号则转＄DFF0，继续进行上述运算；若没有括号则结束第一类运算。

对于本例，当括号运算完成后，有关单元的内容如下：

＄7600＝41	＄7601＝80
＄7602＝20	＄7603＝20
＄7604＝20	＄7605＝20
＄7606＝20	＄7607＝20
＄7608＝20	＄7609＝20
＄760A＝20	＄760B＝20
＄760C＝20	＄760D＝20
＄760E＝42	＄760F＝B6
＄7610＝33	＄7611＝32
＄7612＝35	＄7613＝B5
＄7614＝31	＄7615＝2E
＄7616＝32	＄7617＝2D
＄7618＝36	＄7619＝20

$420=30	$421=35
$422=36	$423=37
$424=39	$425=33
$426=38	$427=35
$428=38	$429=00
$42A=20	

$38C=30	$38D=35
$38E=36	$38F=37
$390=39	$391=33
$392=38	$393=35
$394=38	$395=00
$396=20	

六、除法运算过程

1、若算式中“÷”是第一个算符则从$7600以后单元中取出被除数送入$400～$40A中，送数时要删去小数点，把小数点后的位数送入$409单元；若不是第一个算符，则从$430～$43A中取出上一次的运算结果送入$400～$40A中。对于本例，因括号后紧跟“÷”运算符，所以要从括号运算结果暂存器$38C～$396中取出数据送入$400～$40A，同时令$7600=H20、$7601=H20、760E=H20。被除数取出后，要使其最后一位数存在$408单元中，其余各位依次向前存放。

2、取除数存入$410～$41A，存数要求同被除数。

3、每当从$76××单元中取出数据时，要随时将其置入H20。

4、调$EA08子程序，将$400～$408、$410～$～$418单元的ASCII码还原为数字。

5、检查$410是否为0，若$410=0则调$EB60子程序令除数各位向前移动一个单元。每调一次检查一次$410，只要$410=0就调$EB60，直到$410≠0为止，由$0B单元记录除数的移位次数。

6、检查$400是否为0，若$400=0则调$EB4A使被除数各位前移一个单元，也要反复调用，直到$400≠0为止。用$0A单元记录被除数的移位次数。

7、比较被除数与除数相同位数数据的大小：若被除数大则调$E776进行一次以减代除的减运算。所减除数的次数由$9F记录，也要反复调用，每减一次，比较一次两数的大小，直到不够减为止。

若除数大，则令$8E～$9F存商单元前移一个单元，令除数各位后移一个单元(相当于笔算除法中，商后移一位)。再重复“7”的操作。

8、通过减运算，被除数小于除数时，表示一位商运算完毕，则令$8E～$9F的各位前移一个单元，令$0A+1，记录商的位数。再令被除数各位前移一个单元重复“7”的操作。

9、直到$0A=8时表示商已达到八位有效数字，则调$E989子程序，根据$0A、$0B的取值决定商中小数点后的位数送入运算结果小数标志单元$0A。令$0A=$0A－$0B－1送入$439。

10、调$EB10把存于$97～$9F中的商转换成ASCII码存入$430～$438单元中。

11、调$E9CC子程序检查$430，以判断商是否溢出。

12、调$EA8B子程序对商进行整形操作，删去小数点后面的0，确定小数点的位置，把

H2E 送入小数点存储单元＄43×。除法运算结束。

对于例题，“÷”运算结束后各单元内容如下：

＄7600～＄7612＝20

＄7613＝B5	＄7614＝31
＄7615＝2E	＄7616＝32
＄7617＝2D	＄7618＝36
＄7619＝20	
＄430＝31	＄431＝37
＄432＝34	＄433＝37
＄434＝35	＄435＝30
＄436＝2E	＄437＝33
＄438＝33	＄439＝02
＄43A＝20	

七、乘法运算过程

1、若“×”是算式中第一算符则从＄76××单元取出被乘数送入＄400～＄408中，并根据是否有小数点对＄409置入小数的位数。若不是第一运算符则把上一次的运算结果从＄430～＄43A或＄420～＄42A中取出送入＄400～＄40A中，并使被乘数的最后一位存入＄408单元。

2、取乘数存入＄410～＄41A中，使乘数的最后一位数存入＄418单元。

3、每当从＄76××单元取出数据时，要随时置入H20。

4、调＄EA08子程序，把＄400～＄408、＄410～＄418中的数据ASCII码还原为单位数字。

5、调＄EC3C子程序决定乘积的符号送入＄43A；调＄EADF子程序检查被乘数是否为0；调＄EAEC子程序检查乘数是否为0。

6、调＄E6FD子程序进行以加代乘的乘法运算。运算过程如下：

(1) 以＄401单元为控制器，累加＄410～＄418中的各位，累加之和存入＄8E～＄9F中，从最后一个单元向上存放。同时，将和转换成ASCII码存入＄430～＄438。每加一次令＄401减一，当＄401＝0时第一位乘运算结束。

(2) 令＄8E～＄9F、＄400～＄408分别前移一个单元，以便进行下一位乘积运算，并令＄0A＋1，记录乘运算的位数。

(3) 前移后检查＄401单元，若＄401≠0则转(1)，进行下一位乘运算；若＄401＝0则转(2)。

(4) 当＄0A＝8、即乘运算达到八位时，结束运算，进行乘积处理。首先检查存于＄8E～＄9F中乘积的位数，若超过八位时则检查＄409、＄419单元，看它们是否为0，若不为0(表示进行的是小数乘运算)，先由＄409控制＄8E～＄9F的内容后移一个单元，每后移一次令＄409减一。若＄409＝0后，乘积仍多于八位，再由＄419控制令＄8E～＄9F后移，直到乘积等于八位为止。以上移位操作的目的是自动删去乘积中超过八位的小数部分。若＄409＝0、＄419＝0、而乘积仍多于八位则进行溢出处理。

7、把＄97～＄9F中的乘积转换成ASCII码，送入＄430～＄438中，本次乘运算结束。再检查以后的算式中是否仍有“×、÷”运算，若有则转入相应运算；若没有则结束本类运算。

对于示例，乘运算结束后，有关单元内容如下：

$ 7600～$ 7616＝20

$ 7617＝2D	$ 7618＝36
$ 7619＝20	

$ 430＝32	$ 431＝30
$ 432＝39	$ 433＝37
$ 434＝30	$ 435＝30
$ 436＝2E	$ 437＝33
$ 438＝39	$ 439＝02
$ 43A＝20	

八、减法运算

1、取数过程同乘法运算。

2、根据 $ 409、$ 419 的数值，令 $ 400～ $ 408、$ 410～ $ 418 的各位移位，使两数的个位数对齐。

3、根据 $ 40A、$ 41A 决定运算结果的符号存入 $ 43A。

4、调 $ EBE6 子程序比较两数的大小。若被减数小于减数则调 $ EBC6，令被减数与减数互相交换位置，使绝对值大的数存入 $ 400～ $ 408 中。

5、令 $ 400～ $ 408 对应减 $ 410～ $ 418，所得之差转换为 ASCII 码存入 $ 430～ $ 438。

6、调 $ EA8B 子程序对差整形，加入小数点，减运算结束。

通过以上四步运算，例题计算结束，各单元内容为：

$ 7600～ $ 7619＝20

$ 430＝32	$ 431＝30
$ 432＝39	$ 433＝36
$ 434＝39	$ 435＝34
$ 436＝2E	$ 437＝33
$ 438＝39	$ 439＝02
$ 43A＝20	

即　　(3982＋56789876)÷325×1.2－6≈209694.39

九、加法运算过程

1、取数过程同乘法运算

2、调 $ EA52 子程序判断小数点位置，使两数的个位对齐，并根据 $ 409、$ 419 单元的内容计算小数位数存入 $ 439。

3、调 $ EBFD 子程序判断符号位，根据 $ 40A、$ 41A 的数值确定和的符号存入 $ 43A。

4、调 $ E6FD 子程序进行对应位相加运算，相加之和存入 $ 97～ $ 9F 单元。同时将其转换成 ASCII 码存入 $ 430～ $ 438 中。

5、调 $ EA8B 子程序对结果整形，加入小数点，结束加法运算。

十、运算结束处理过程

当整个算式运算结束后，要把运算结果由 $ 430～ $ 438 中取出送入 $ 3F0～ $ 3F8 中，以备送显。

结果转存处理程序在$DDA9~$DE00。处理过程如下:

首先检查$43A,若$43A=H2D则把H2D送入$3F0;

然后从$430开始依次向后查找,一直找到不等H30的单元,再判断是否是H2E。若不是H2E则把以后各单元内容送入$3F0以后的单元;若是H2E则向前查一个单元,若是H30则直接把小数点以后各单元的内容送入$3F0以后;

最后再置入一个H0D,作为送显结果的结束标志。

令$D8=1,置运算结束标志。运算过程结束。

运算过程小结

1、算式运算仍遵循先括号内、后括号外、先乘除、后加减的运算规则,先从第一类算起,只有本类运算结束再转入下一类运算。

2、在进行加减运算时,要使小数点位对齐。

3、在进行减运算时,要把绝对值大的数值放入$400~$408。

4、在进行乘运算时,两数存入$400~$408、$410~$418后,要使数据的最后一位存入$408、$418,其它各位依次向前存放。

5、在除运算中,两数存入$400~$408、$410~$418时要使两数据的第一位分别存入$400、$410,其它各位依次向后存放。

6、每次运算完毕都要检查结果的溢出情况和对结果整形,删去小数部分最后的0,加入小数点,恢复送显结果的原形,以备送显。

3·4·5 运算结果的输出过程

运算结果的输出送显仍在中断处理中完成,共分三步进行:

一、检索送显PPU的地址指针

运算结束后置$D8=1,当执行中断处理程序$DC77时转$DCC7,通过调用$DE31子程序检索出结果送显的PPU地址变址指针存入$B6单元,把PPU地址置入$33、$35中。最后令$D8=2,置送显地址检索结束标志。

二、送显结果

检索到PPU送显地址后,在下一次中断期间,由$D8=2调$DCF9子程序,首先以$DA为变址从$3F0开始取出一个字符存入$DD单元;然后调$F3F8子程序设置三页送显字符框架,调$F210子程序调整PPU地址,调$F115子程序调整羽毛笔的位置;第三,把$DD中保存的结果字符送入$304,以便在下一次中断送显。同时以$B6变址把结果字符存入$63D以后的单元;最后置$FF=8发出送显结果字符时的"啾"声。"啾"声的发声延时时间由$21=8控制,发声时间为0.16秒。

当在结果的暂存单元$3F0~$3F8中读到H0D时,表示结果送显完毕,把羽毛笔坐标调整到结果最后一位的下一个位置,向$63D+$B6中送入空格(H20)。令$D8=3,置结果送显结束标志。并令$FF=40发出送显完毕后的一长串"嘟噜"声。

三、令羽毛笔与算式间隔一个空格

运算结果送显完毕,由$D8=3调$DD75,检查$63D+$B6单元的内容,若等于H20则令$D8=4返回;否则向该单元置入H20。至此,结果送显操作完毕。

四、返回算式输入状态

由$D8=4调$F50F子程序,令羽毛笔在新的坐标位置显示。令$D8=0恢复输入算式

的等待状态。准备接收新的算式输入。

3·4·6 CAL·板屏幕编辑过程

CAL·板的屏幕编辑包括使用四个方向键控制羽毛笔上、下、左、右移动；使用"Del"键删除字符；使用"Ins"键插入空格；使用"Home+Shift "键清屏；使用"Home"键把羽毛笔移至屏幕左上角等过程。下面分别介绍。

一、部分清屏与只按"Home"键

CAL·板的清屏操作，实质上是部分清屏。它只清除操作者输入的算式部分和运算结果部分，而CAL·板的画面则完全保留。

当按"Home+Shift"键时，$37=0C，由$DC77转$DCF5置清屏标志$7D=1。在下一次中断期间由$7D=1转$DCE1清屏处理程序。清屏操作过程如下：

1、调$D197部分清屏子程序，由$7D控制逐行清屏。

首先由$D1A3～$D1BB设置第一行清屏框架为：

$301=20

$302=E3

$303=5A

$304=20

$305=00

清除第一行，并将PPU地址高位置入$7C。

然后调$EEFB子程序令$63D～$7D3=H20，即清除CAL·板工作区。

第三、令$7D加一，记录清屏的行次。

第四、调$F50F子程序把羽毛笔移到屏幕左上角输入算式的第一行首位。

以上四步完成了第一行的清除。

2、在下一次中断期间仍调$D197转$D1E8，令清屏框架中$302的内容加H20(使PPU的地址指向下一行算式)、令$301的内容加进位"C"，再令$7D加一，进行清除下一行。

3、以后六行的清除过程与2相同，直到$7D=H0A时说明清屏操作完毕，令$7D=0，结束清屏。

当按"Home"键时对$7D置入HFF，则只执行调用一次$EF0D和$F50F两个子程序，把羽毛笔移到首行首位置后，令$7D=0，结束操作。

二、用方向键移动羽毛笔

移动羽毛笔和删除字符及插入空格的操作都在按键综合处理程序$EF2C中完成。其工作过程如下：

1、右移羽毛笔

假定右移前羽毛笔位于第一行首，则它的X坐标$2C3=H22、Y坐标$2C0=H3C($30=H22、$31=H3C)，对应的PPU显示地址为$20E4($32=H20、$34=HE4)。

当按"→"键时，$37=H1C，则执行$EF97～$EF9F程序段。

首先由$EF97调$F210子程序令$34加一，使PPU地址指向$20E5。

然后调$F115子程序令$2F加一、$30加8，使羽毛笔的X坐标后移一个字符位置。

最后转$F090调$F50F子程序，令羽毛笔在新的坐标上显示。从而完成了羽毛笔右移一个字符的操作。

如果连续右移，则重复以上过程，当羽毛笔右移到右边界时，则在调＄F115 的过程中完成换行的操作。此处不多赘述。

2、下移羽毛笔

当按“↓”键时，＄37＝H1F，则进入＄EFC7～＄EFD8 程序段。进行羽毛笔下移的操作。

首先调＄F259 子程序，令＄34＋H40、＄32＋HC，即令 PPU 送显字符地址指向＄2123(假定开始羽毛笔位于第一行首)，反映在屏幕上就是羽毛笔下降两显示行(跳过划横线的一行)，在 CAL·板则为下降一个算式行。

然后调＄F1DE 子程序，令＄2F＝＄2F＋H18、＄2E＝＄2E＋HC，即相当于输入了 24 个空格。令＄31＝＄31＋H10，即令羽毛笔的 Y 坐标下移两个字符的位置。

最后转＄F090 令羽毛笔在新的坐标上显示。

若连续按“↓”键则连续下移，当＄31＝HAC 时则保持原值不再改变，反映在屏幕上是羽毛笔移到算式第八行后不再下移。

有关羽毛笔的上移和左移工作过程则是下移和右移的逆操作，控制程序分别在＄EFB1～＄EFC2 和＄EFA4～＄EFAC。留给读者分析。

三、删除字符

当欲删除羽毛笔前面的一个字符时可按“Del”键，这时＄37＝08，由按键综合处理程序＄EF2C 查出“Del”键的处理程序在＄F007～＄F019。

假定我们在第一行已输入了数字“1023.45”，我们欲将其改为“123.45”，则要先用左移键把羽毛笔移到“3”的位置，然后再按“Del”键。

按“Del”键前，有关单元的内容为：

＄30＝32	＄63D＝31
＄31＝3C	＄63E＝30
＄32＝20	＄63F＝32
＄34＝E6	＄640＝33
＄2F＝3	＄641＝2E
	＄642＝34
	＄643＝35
	＄644＝20

按“Del”键后，首先调＄F473 子程序，检索出本行中羽毛笔后面的最后一个字符的 PPU 地址低位存于＄35 中，对上例有＄35＝HEA，并对＄644 置入 H20。

然后调＄F2A9 子程序把羽毛笔以后直到行尾的各字符前移一个位置。前移过程是：

1、令＄2F－1，＄2F＝2，把＄2F 的值存入＄09 单元，＄2E 的值存入＄0A。

2、令＄34－1，＄34＝HE5，以 PPU 的地址＄20E5 为基础设置三页送显框架，先对＄304 置入 H20。

3、把＄34 单元的内容暂存＄08、＄32 的内容暂存＄07，令 X＝1。

4、令 Y＝＄2F，从＄63E＋Y 中取出数据送入＄303＋X，同时送入＄63D＋Y。实质就是把＄63E 以后各单元的内容前移一个单元。

5、令 X＋1、＄2F＋1、＄34＋1，重复 4、5 项的操作。

6、直到＄34＝HFC 时，即取数到本行的行尾则结束数据转移处理。

7、向＄303 中送入 X 寄存器的数值，即＄303＝＄2F－＄09＝H17，令＄31B＝0，置本次送

显结束标志。

8、从＄07、＄08 中取回羽毛笔当前位置的坐标值，即＄32＝＄07＝H20、＄34＝＄08＝HE5；从＄09、＄0A 单元中取回字符计数器的当前值，即＄2F＝＄09＝2、＄2E＝＄0A＝0。转＄F090 令羽毛笔在新的坐标位置显示，反映在屏幕上就是令羽毛笔及其后的各字符前移一位。

对上例，按一次"Del"键后，有关单元的数据如下：

＄300＝1B	＄63D＝31
＄301＝20	＄63E＝32
＄302＝E5	＄63F＝33
＄303＝17	＄640＝2E
＄304＝32	＄641＝34
＄305＝33	＄642＝35
＄306＝2E	＄643＝20
＄307＝34	＄644＝20
＄308＝35	＄645＝20
＄309＝20	＄646＝20
……	……
＄31A＝20	
＄31B＝00	

关于按"Ins"键插入空格的工作过程，实质上是"Del"键的逆过程，处理程序在＄F01E～＄F030。留给读者分析。

3·4·7 CAL·板的退出操作

当 CAL·板运算结束，欲退出该板而进入其它工作板时则按"Esc"键。退出过程如下：

一、退出 CAL·板

按"Esc"键后＄37＝H1B。由中断程序＄DC77 控制进行以下操作：

1、调＄F4B1 子程序，重新定义八个功能键为 C1～C8。

2、调＄DF9A 子程序，把存于＄63D～＄714 中的计算板输入的内容送入 CAL·板数据库＄7100～＄71D7 保存。

3、令＄703A＝H5A、＄703B＝HCC，置运算板结束标志。

4、令＄FC＝0，关背景音乐伴奏。

5、令＄FE＝1，发出换板声响。

6、令＄60＝0，关闭运算板工作程序入口。

二、复绘对话板画面

1、令＄60＝9，转＄CF06 进行清屏、清二页。

2、令＄60＝A 转＄D515，定义对话板中的各 SP 块。

3、令＄60＝8、＄6B＝1，绘制对话板画面。

4、令＄60＝6、＄C3＝0D，转＄D2A4 清两栏，送显

"CALCULATOR" FINISHED

5、令＄60＝6、＄C3＝H10，送显

NEXT SELECT

6、令＄60＝6、＄C3＝8，送显

"MESSBOARD"

NEED?

7、令＄60＝H0C、＄6B＝3，转＄D44A，显示四个提示栏中的提示。

8、令＄60＝H15，转＄D06E 进入等待操作者选择输入状态。

CAL·板工作小结

1、CAL·板的工作分两步进行：算式的输入、运算结果的输出及屏幕编辑等在中断程序＄DC77 中进行；算式的合法性检查和运算在复位主程序进行。这样安排的目的是：当算式较复杂、运算时间较长时，避免中断处理占用机时太多出现画面闪烁。

2、CAL·板可以进行浮点数运算。在运算处理中使用的 RAM 单元较集中（＄38C～＄43A、＄8E～＄9F），因此只要处理好有关单元的保护，把算式置入＄7600～＄76FF，调用有关运算子程序就可在 F BASIC 状态下进行浮点数的运算，以弥补 F BASIC 没有小数运算的不足。

3、通过 CAL·板的运算过程可以看出，运算数暂存单元、运算结果暂存单元中还有空余单元，如第一运算数仅使用了＄400～＄40A 共 11 个单元，还有＄40B～＄40F 共 5 个单元没有使用。因此，只要把运算程序中有关数据稍加改变就可以得到具有 13 位有效数字的运算程序，可使运算精度提高五个数量级，这无疑为系统软件的开发提供了一条捷径。

4、CAL·板工作过程中的音响汇总于表 3－25。

表 3－25　CAL·板音响统计表

音响控制单元	音　响　效　果	使　　用　　场　　合
＄FE＝40	"发啦"声	输入算式字符时
＄FF＝8	"啾……"声	送显结果字符时
＄FF＝40	一长串嘟噜声	送显结果完毕时
＄FC＝1	背景音乐伴奏	伴随 CAL·板的工作全过程

有关音响的发声过程，留待后文讨论。

3·5　打字板工作过程分析

MES·板有以下特点：

1、由于电脑游戏机键盘键位按英文打字机标准键盘的键位排列，因此可以在 MES·板进行英文打字练习，以熟悉键位，提高程序的输入速度。

2、在小写字母状态下（羽毛笔为绿色时）使用数字键、算符键、配合 Alt 键可进行制表练习，方便地设计出各种表格。

3、MES·板具有 CAL·板相同的屏幕编辑能力，因此可方便地使用编辑键，进行各种屏幕编辑。

4、屏幕下方有不停变换种类的卡通从右向左运动，还有不同于 CAL·板旋律的背景音乐伴奏，从而使 MES·板的魅力比 CAL·又增一筹。

3·5·1　MES·板工作状态的建立

一、MES·板的进入

在对话询问中，当询问

"MESSBOARD" NEED?

时按 F1 键或在询问其它板时键入"MES·"回车后都可进入 MES·板。操作过程如下：

1、清提示栏、清两栏、送显"OK!"。延时、清两栏、送显

"MESSBOARD" START.

2、令＄60=9、查表 3—4 转＄CF06，清屏、清二页。

3、令＄60=8、＄6B=4、转＄CEF8，令＄8A=＄6B=4，在下一次中断调＄EE43，对 MES·板各配色单元置入配色，送显

"MESSAGE BOARD"。

4、令＄60=8、＄6B=7，转＄CEF8，令＄8A=＄6B=7，在下一次中断期间调＄EE43 绘制 MES·板的边框及墨水瓶。MES·板的画面数据与图 3—5、图 3—6 基本相同，区别是把图 3—5 中的横线、提示删去，把板头由"CALCULATOR BOARD"换为"MESSAGE BOARD"。MES·板的 PPU 配色区颜色代码见表 3—26。

表 3—26　MES·板 PPU 配色区颜色代码一览表

PPU 配色地址	颜色代码	PPU 配色地址	颜色代码
3F00	0F	3F0A	36
3F01	30	3F0B	07
3F02	27	3F0C	0F
3F03	07	3F0D	30
3F04	0F	3F0E	2C
3F05	30	3F0F	02
3F06	36	3F10	0F
3F07	15	3F11	36
3F08	0F	3F12	16
3F09	30	3F13	02

5、令 60=H1D，转 D124，令＄63D～＄7D4=H20，把＄7400～＄7597 的内容送入＄63D～＄7D4，即把 MES·板数据库中的内容取回 MES·板(第一次使用 MES·板时，＄7400～＄7597=H20)。

6、令＄FC=2 启动背景音乐开始伴奏。

7、令＄FE=1 发出进入 MES·板的一长串"嘟噜"声，并令＄703A=1、＄703B=1，置 MES·板工作标志。

8、对＄2C0～＄2DF 赋值，定义向上羽毛笔，初始状态数据同 CAL·板。令＄60=0。

9、令 60=H0B，转＄DB97 进入 MES·板工作状态。

3·5·2　MES·板工作程序简介

MES·板的工作过程全部在中断处理中完成，控制程序位于 ROM 区＄DB2F～＄DC62。其中：

- ＄DB2F～＄DB96 为数据区
- ＄DB97～＄DBCC 为主控程序
- ＄DBCD～＄DBE1 为 MES·板退出处理程序
- ＄DBE2～＄DC62 为 MES·板卡通运动控制程序

由于 MES・板的屏幕编辑操作调用了公用部分清屏子程序 $D197 和综合按键处理程序 $EF2C，所以 MES・板的工作程序极为简洁，主控程序仅占用了 54 个内存单元。

因 MES・板的屏幕编辑过程和字符输入过程与 CAL・板基本相似，所以对于字符的输入、羽毛笔画圈、字符的送显等一系列操作过程的分析从略。重点讨论卡通运动和种类变换及改变配色的控制过程，并简要介绍制表工作过程。

3・5・3 运动卡通的控制过程

MES・板的工作过程中，在背景音乐的伴奏下，有八种卡通轮流以不同的色彩在屏幕下方运动，周尔复始，永不停止。从而使枯燥的字符输入过程变得活泼有趣。因此，了解 MES・板对运动卡通的控制必然会对我们在编写游戏程序中、对角色以外卡通的自动运动控制提供极有价值的参考，甚至可以直搬照用。这无疑给我们编写游戏程序提供了一条捷径。

一、卡通控制程序简介

卡通控制程序为 $DBE2～$DC62，按功能可大体分为四段：

1、$DBE2～$DC10 参量赋值子程序

通过该段程序，对运动卡通的控制参量单元 $B2、$B4、$B6、$28 赋值，以便进行卡通种类的选取、显示坐标的变换、配色的调整等操作。

2、$DC11～$DC28 选取卡通的配色、种类处理程序

该段程序的作用是：根据 $B3 的取值、通过对 $B5 赋值，控制卡通种类的调换和配色的选取，使卡通的色彩发生变化。

3、$DC29～$DC4D 卡通运动处理程序

该段程序的作用是：根据 $B2、$B3、$B5 的取值，选择卡通序号以改变显示卡通的姿势，改变显示卡通的坐标以产生运动效果。

4、$DC4E～$DC62 卡通的种类、配色选定处理子程序

该子程序由 $DC11 种类、配色处理程序所调用，以实现 $DC11 的控制功能。

二、重要工作单元介绍

在分析讨论前，先熟悉掌握卡通运动控制的重要工作单元的作用，将方便程序的讨论，利于领会理解。

1、$B6：卡通运动总量（运动时间）控制器。只有 $B6 不为 0 时卡通才运动。仅当有字符输入后 $B6 才不为 0。一般输入一个字符可令 $B6 从 0 增加至 H0F。如不断地输入字符则 $B6 恒大于 0。

2、$28：卡通运动量控制器。每次赋值 $28＝H10，每执行一次中断 $28 单元减一。只要 $B6≠0，当 $28＝0 时则立即赋值，再令 $28＝H10。$28 每赋值一次，可控制卡通前进四点。

3、$B4：卡通运动速度控制器。每一次赋值为 $B4＝8，每执行一次中断 $B4－1，仅当

$B4 AND 3＝0

时才令卡通前进一点。所以 $B4 每次赋值，卡通前进两点。只要 $28≠0，每当 $B4＝0 则立即赋值，令 $B4＝8。由于取 $B4 和 3 相与为 0 的间隔调整卡通的坐标，所以称 $B4 为速度控制器。

4、$B2：卡通序号调整器。为使卡通运动姿势形象，须在改变卡通显示坐标的同时，改变卡通的显示姿势。MES・板规定：卡通每前进 2 点改变一种姿势；每前进一步改变三种姿势。以马里欧哥哥为例，马里欧每走一步，令马里欧走 1、马里欧走 2、马里欧走 3 三种姿势都显示

一次；规定马里欧每前进 8 点为一步。

每当 $B4=0 时令 $B2+1，并令

$B2= $B2 AND 3

因而控制了 $B2 的取值为 0～3。由 $B2 的取值决定选取卡通的姿势类别，因 $B2 从 0～3 变化，所以卡通运动的姿势也不停地反复变化。

5、$B3：卡通运动产生器。它的取值映射于卡通的 X 坐标。每当

$B4 AND 3=0

时，令 $B3－1，把 $B3 的值送入卡通的 X 坐标单元，使卡通在新的坐标上显示。由于 $B3 的取值为 HFF～0，所以卡通运动方向为从右向左。

6、$B5：卡通种类选择器。当一类卡通从右向左运动到左边界、即 $B3=0 时，则令 $B5 +1，并令

$B5= $B5 AND 7

把 $B5 的取值范围限制在 0～7 之间，由 $B5 控制选择运动卡通的种类。所以 MES·板中只有八类卡通反复运动。

三、运动卡通的控制过程

1、对卡通运动量控制器赋值

卡通的运动量由 $B6、$28 两个单元决定，$B6 为运动总量控制。

在进入 MES·板后，当未按任何键时，MES·板主控程序不起作用，故这时无卡通运动。当按任一键后，$37≠0，随后使 $39≠0。仅当 $39≠0 时，才在执行主控程序 $DB97 调 $DBE2 子程序中令 $B6+1，使 $B6=1。由于 $B6=1，在执行 $DC06 时对 $28 赋值，使 $28=H10。

2、对 $B4、$B2 赋值以决定卡通的运动速度和组成卡通的 4 块 SP 块的序号。

当 $28≠0 后，在调 $DBE2 子程序时，首先对 $B4 赋值，令 $B4=8；然后令 $B2+1；最后令 $B4－1，检查 $B4 与 3 相与的结果，若不等于 0 则结束中断处理。

3、对卡通运动产生器 $B3 赋值

仅当 $B4 AND 3=0 时才转 $DC11，令 $B3－1，使 $B3=HFF，启动卡通运动。

4、卡通种类的选取和定义

卡通种类由 $B5 的取值决定。因 $B5 的初值为 0，故第一个运动的卡通必然是马里欧哥哥。

卡通种类的选取和卡通序号的选定由 $DC29～$DC4D 程序段完成。工作过程如下：

(1) 执行 $DC29、令 $00= $B3=HFF，设置定义卡通的 X 坐标；令 $01=HBF、设置定义卡通的 Y 坐标。

(2) 以 $B5 为变址指针，从 $DB4F～$DB5E 数据区中取出 SP 块序号地址存入 $03、$04 单元中。由于 $B5 的取值范围是 0～7，所以与 $B5 对应有 8 个卡通序号地址。它们的对应关系见表 3－27。

开始时 $B5=0，由表 3－27 可查出 $03=H2F、$04=HDB。

(3) 以 $B2 为变址指针，以表 3－27 的卡通种类序号地址为基址，查出组成卡通的第一块 SP 块的序号存入 $02 单元中。

因 $B2 的取值为 0～3，所以每类 SP 有四种初始序号供选择。$B2 与 SP 序号的对应关系见表 3－28。

表 3—27　$B5 与 SP 种类、SP 序号地址对照表

$B5 的取值	卡通种类	卡通序号地址	$B5 的取值	卡通种类	卡通序号地址
0	马里欧	DB2F	4	丽莎	DB3F
1	飞鱼	DB33	5	乌龟	DB43
2	妖怪	DB37	6	螃蟹	DB47
3	企鹅	DB3B	7	飞鸟	DB4B

表 3—28　$B2 与 SP 种类序号对照表

$B5	马里欧	飞鱼	妖怪	企鹅	丽莎	乌龟	螃蟹	飞鸟
0	00	40	58	60	1C	B8	C0	C8
1	04	44	5C	64	20	BC	C4	CC
2	08	40	58	60	24	B8	C0	C8
3	04	44	5C	64	1C	BC	C4	CC

开始时，$B2=1，所以玛丽的首序号为 4。然后令 $03=22，决定运动卡通由 4 块 SP 块组成，SP 块的排列方式两层、每层两块。

(4) 转 $ECBE 定义卡通子程序，定义第一个运动卡通。经 $ECBE 子程序处理后，定义的卡通数据如下：

$200=BF　　$204=BF
$201=04　　$205=05
$202=22　　$206=22
$203=FF　　$207=07

$208=C7　　$20C=C7
$209=06　　$20D=07
$20A=22　　$20E=22
$20B=FF　　$20F=07

5、SP 运动速度的控制过程

(1) 从 $DBEC 以后的程序中可以看到，当 $B4≠0 时则不改变 $B2 的数值，仅令 $B4－1。仅当

$B4 AND 3=0

时(即 $B4=0 或 $B4=4 时)转 $DC11 令卡通前进一点。所以 $B4 每赋值一次，SP 以一种方式前进两点。

(2) 当 $B4=0 时，则对 $B4 重新赋值，同时令 $B2+1，即 $B2=2，当 $B4 递减到 4 时转 $DC11，取下一组序号定义 SP。这时定义 SP 的数据如下：

$200=BF　　$204=BF
$201=08　　$205=09
$202=22　　$206=22
$203=FD　　$207=05

$208=C7　　$20C=C7
$209=0A　　$20D=0B
$20A=22　　$20E=22

$20B=FD　　　　$20F=05

可见，$B4 每赋值一次，可令 SP 前进两点；SP 每前进两点则令 $B2+1，改变一次 SP 的姿势。从而使卡通产生运动效果。

如果欲使卡通的运动速度加快或减慢，我们可以调整 $DBFD～$DC04 中与 $B4 相与的数值。显然，若改成

$B4 AND 2=0

则将使运动速度提高一倍；相反若取

$B4 AND 7=0

则将使速度减慢一倍。

6、卡通种类与配色的改变过程

当马里欧运动 255 点以后，$B3=1，当执行 $DC11 时，$B3-1=0。这时执行 $DC1C～$DC28 程序段，首先令 $B5+1=1，由表 3-28 可知，这时选择的 SP 是飞鱼；然后调 $DC4E 子程序令 $00=H18、$01=H3F、置 PPU 首址 $3F18；第三，以 $B5 为变址指针，从配色地址数据区 $DB87～$DB96 中取出配色地址置入 $02、$03 单元；最后转 $ED65 把配色地址及颜色代码送入三页，以便在下一次中断时送入 PPU 的配色单元。

由于 $B5 取值范围为 0～7，所以共有 8 个配色地址和 8 组配色代码。它们的对应关系见表 3-29。

表 3-29　$B5 与 PPU 配色代码对照表

$B5	配色代码地址	送数指南	颜色代码			
			3F18	3F19	3F1A	3F1B
0	DB5F	4	0F	36	16	02
1	DB64	14	0F	30	15	12
2	DB69	14	0F	30	12	16
3	DB6E	14	0F	30	26	12
4	DB73	14	0F	35	25	17
5	DB78	14	0F	30	26	19
6	DB7D	14	0F	30	02	16
7	DB82	14	0F	30	26	12

由表 3-29 可知，第一次改变配色时三页数据如下：

$301=3F　　　　$302=18

$303=04　　　　$304=0F

$305=30　　　　$306=12

$307=16　　　　$308=00

7、卡通的连续运动控制

(1) 从 $DBE2 程序段可见，当我们每输入一个字符时，羽毛笔要经过 15 个中断周期才画圈完毕，因此可令 $B6 从 0 增至 H0F。以后再有字符输入时，$B6 将继续增加。由此可见，每输入一个字符，$B6 最少增加 H0F。虽然 $28 赋值一次要令 $B6 减一，但要 16 个中断周期才减一次。因此 $B6 的增加速度是极快的。

(2) 由 $DC06～$DC10 程序段可以看出，每当 $28=0 以后，只要 $B6≠0 则反复对 $28 单元赋值。

(3) 由 $DBEC～$DBF3 程序段可见，每当 $B4=0 以后，只要 $28≠0 则反复对 $B4 赋

值。

归纳以上三点可得出如下结论，$B6、$28、$B4 三个单元和 SP 运动的关系是：

$B6 衍生 $28

$28 衍生 $B4

$B4 控制 SP 运动

它们之间的数量关系是：

每对 $28 赋值一次则对 $B4 赋值两次

每对 $B4 赋值一次则令 SP 运动两点。

也就是说 $28 单元赋值一次，SP 运动四点；输入一个字符 $B6 增加 H0F。因此，每输入一个字符可令 SP 前进 60 点。当我们不停顿地输入字符时则可使 SP 周尔复始、连续运动。

3·5·4 制表工作过程分析

在 MES·板，当按"CapsLock"键、处于小写字母状态时，按"Alt"和数字键可调用 13 种图形制表。

制表工作分两步进行。第一步在读键程序中读出图形的 ASCII 码存入 $37 单元；第二步在 MES·板处理程序 $DB97 中把图形送入 PPU 显示。可见，制表的关键在读键处理中。我们知道，电脑游戏机的键盘上没有与制表图形对应的按键，因此欲使用按键调出图形，必须对按键的功能定义进行特殊处理。下面我们重点讨论这一过程。

一、制表状态的进入

1、进入制表状态，首先要按"CapsLock"键进入小写字母状态。我们知道，"CapsLock"键的键值存于 $634 单元中，因此有 $634=8。当读键程序调用 $B02C 时，则令 $6E=0、$6D=3。由键码数据区 $B0EF 读出键值为 HFF，因而转 $B090，令 $72=FF、$37=0、结束读键。

2、在按键综合处理程序 $EF2C 中，因 $37=0 故转 $F090 调羽毛笔定义子程序 $F50F，再调 $D18C 令 $00=H42、$01=HCA、转 $EC86，把羽毛笔的 8 个 SP 块的状态单元置入 01，使羽毛笔显示绿色，从而进入制表工作状态。

二、制表图形的读入

进入制表状态后，按 Alt+数字键（或算符键）就可以调入图形制表。调入过程如下：

以"Alt+1"为例

1、按"Alt+1"键，查表 3-1 有 $63B=6（按"Alt"键有 $63B=4、按数字"1"键有 $63B=2，两键相加有 $63B=6）。

$63B AND 4≠0

故由 $B001 程序控制，转 $B057。

2、令 $71=HFF，置按"Alt"键标志，转 $B024，令

$63B= $63B AND HE3

（屏蔽 $63B 的 D2、D3、D4 位）。

3、由 $B02C 和 $B06D 处理，令 $6E=7、$6D=1。因 X 的偏移量为 H39，故由 $71=HFF 控制转 $B0BA，查出"Alt+1"的键码为 HE8，送入 $37。图形读入过程结束。

通过以上讨论，我们找出了制表图形的数据区为 $B20C～$B24C。为方便查用，特把各制表键对应的键值及键码列于表 3-30。

表 3—30　制表状态、按键与图形对照表

Alt＋按键	键值存放单元	键值	$ 6E	$ 6D	X 寄存器	键码地址	键码	图形
1	63B	2	7	1	39	B245	E8	┼
2	63B	1	7	0	38	B244	E9	┴
3	63A	1	6	0	30	B23C	EA	┬
4	639	1	5	0	28	B234	EB	┤
5	639	2	5	1	29	B235	EC	├
6	638	1	4	0	20	B22C	ED	─
7	638	2	4	1	21	B22D	EE	│
8	637	1	3	0	18	B224	EF	┌
9	637	2	3	1	19	B225	F0	┐
0	63C	1	8	0	40	B24C	F1	└
—	635	2	1	1	09	B215	F2	┘
＝	635	8	1	3	0B	B217	FD	□
\	634	2	0	1	01	B20D	FE	■

3·5·5　MES·板的退出处理

按"Esc 键"可退出 MES·板，返回对话板询问工作状态。

按"Esc 键"后，$ 37＝H1B，则执行退出 MES·板处理程序。退出过程如下：

1、调 $ DBCD 子程序，把 $ 63D～ $ 83C 的内容送入 $ 7400～ $ 75FF 单元，即把 MES·板输入的内容存入其数据库。

2、转 $ DC9B，令 $ 703A＝H5A、$ 703B＝HCC，置工作板退出标志。

3、令 $ FC＝00，关闭背景音乐伴奏；令 $ FE＝1，发出一长串换板声。

4、令 $ 60＝0，关闭 MES·板工作程序入口。

5、令 $ 62＝4、$ 60＝9、转 $ CF06 清屏、清二页。

6、绘制对话板画面。（过程同退出 CAL·板）

7、送显　"MESS BOARD"
　　　　NOW FINISHED

延时、清两栏。

8、送显　"NEXT SELECT"

延时、清两栏。

9、送显　"MUSIC BOARD"
　　　　NEED?

进入下一板询问选项状态。

MES·板工作小结

1、子程序 $ DBE2 是运动卡通自动控制的处理程序范例，稍加改造就可以引用到我们的游戏程序中。因此，它具有较高的借鉴和参考价值。

2、通过制表键码的读出过程讨论，给我们自定义功能键（扩展功能键数量）提供了有效的途径。

3、MES·板的效果音响数据汇总于表 3—31。发声过程留待下一节讨论。

表 3—31 MES·板的音响数据汇总表

控制单元	发声数据	音响效果	使用场合
$FC	2	背景打击乐伴奏	整个工作过程
$FE	1	一长串嘟噜声	进入 MES·板时
$FE	40	"发啦"声	输入字符时

3·6 对话工作系统的发声过程详析

发声在对话工作系统中起着重要的作用。它不仅能够渲染各板的工作气氛，更重要的是：发出的各种声响提示，标志着各种状态的建立；各种操作的完成；各个阶段的进入和各种功能的实现。使操作者眼观耳听并用，随时了解电脑的工作状况。在同一时间可一边操作电脑一边处理另外的事情，从而提高了操作者的工作效率。

在游戏程序中，发声将更为重要。有人把发声列为电脑游戏的三大要素之一(背景、角色、音响)，确实不过分。《魂斗罗》游戏之所以在众多的电视游戏中独占鳌头，经年不衰，虽然主要取决于它创造力和想象力紧密结合的精彩无匹的背景设计和角色的增强能力，同时也与它那铿锵抑扬的背景音响和世间罕闻的奇妙绝伦的效果音分不开的。后来的电视游戏诸如：《双截龙》、《沙罗曼蛇》、《松鼠大战》、《人间兵器》等节目之所以能够一炮打响，得到电玩迷们的青睐，归根结底就在于它们借鉴了《魂斗罗》的处理方法，又加以创造和提高。由此可见，音响是编好游戏程序的重要一环。分析研究对话系统的发声过程和方法，是我们编好音响程序的又一捷径。

3·6·1 发声控制程序简介

发声控制程序位于 ROM 区 $FC06～$FFF9 和 $8006～$80AC。其中 $FEE0～$FFF9 是效果音、背景音乐的数据区，$8006～$80AC 是发声音调数据区。

按功能可将发声控制程序 $FC06～$FEDF 分为八段，每段程序的主要作用如下：

1、发声主控程序 $FC06～$FC68

主控程序完成发声数据的采集、发声工作单元的复位、发声随机数的产生和发声开关的状态控制，它控制了整个发声系统的工作。

2、三个主声部赋值子程序 $FC69～$FC97

该子程序完成对三个主声部的 12 个工作单元分别赋值，赋值数据主要取自于 $8006～$80AC。

3、音乐板演奏处理程序 $FC98～$FCE2

该段程序完成 MUSIC 板三段音符(上、中、下段)的演奏(对应于三个声部)，是 MUSIC 板的主要工作程序。

4、音色、音量参数采集子程序 $FCE6～$FD28

该子程序完成音色、音量参数的采集，其中 $FCE6～$FD10 为音色、音量数据区。它把每个单音的音量从强到弱分为 27 段，从而控制发出较长的衰减音。

5、第一主声部效果音处理程序 $FD29～$FDB5

第一主声部的效果音主要由 $FF 单元控制发出。因此，对话工作系统中送显字符时的"哒"声、"当"声、"啾"声和送显运算结果完毕后的"嘟噜"声都由该段程序控制发出。

6、由 $FC 控制的第三主声部效果音处理程序 $FDB6～$FE10

该段程序的作用是：由 $FC 单元控制第三主声部发出效果音。复位画面中的"嘟嘟"声（$FC＝H70）、不规则的"滴哒"声（$FC＝H10）和对话板等待操作者输入时发出的由强到弱的"哆哆"声（$FC＝H68）等都由该段程序控制发出。

7、背景音乐伴奏控制程序 $FE14～$FE92

CAL·板、MES·板的背景音乐由该段程序控制发出。它可控制发出三种不同旋律的背景音乐，每一种旋律由三段乐章组成，分别由 $FC＝1、$FC＝2、$FC＝3 控制。当 $FC＝1 时为 CAL·板背景音乐伴奏、$FC＝2 时为 MES·板背景音乐伴奏、$FC＝3 时的背景音乐没有使用。

8、由 $FE 控制的第三主声部效果音处理程序 $FE93～$FEDF

该段程序通过 $FE 单元控制可使第三声部发出一串"嘟噜"声（$FE＝1）；COMPUTER 栏每次送显完毕后的"叽哩咕噜"声（$FE＝2）；从复位画面进入对话板时的效果音（$FE＝4）；按回车键时的"嘟噜"声（$FE＝H10）和按功能键或输入字符时发出的"发啦"声（$FE＝H40），还有一段小号音色的进行曲（$FE＝H20）等等。

3·6·2 对话工作系统的音响数据汇集与使用的重要单元

一、效果音响

我们知道，对话工作系统中有着不同的音响，标志着各项工作的完成情况。为便于以后讨论，特将音响数据汇总归类列于表 3－32。

表 3－32 对话工作系统中的音响数据汇总表

控制单元	发声数据	音 响 效 果	持 续 时 间	使 用 场 合
FC	01	背景音乐	贯彻始终	CAL·板
FC	02	背景音乐	贯彻始终	MES·板
FC	70	连续的"嘟嘟"声	6.0 秒	复位画面
FC	10	不规则"滴哒"声	12.8 秒	复位画面
FC	70	变调的"嘟嘟"声	3.2 秒	复位画面
FC	68	轻微的"哆哆"声	持续	对话板等待操作者输入
FE	01	一长串"嘟噜"声		换板时
FE	02	"叽哩咕噜"声		电脑栏送显完毕时
FE	04	《七宝奇谋》进小屋		从复位画面进入对话板
FE	10	"嘟噜"声		按回车键时
FE	40	"发啦"声		按功能键或输入字符时
FF	02	"哒哒"声		电脑栏送显字符时
FF	04	"当当"声		操作栏者送显字符时
FF	08	"啾……"声		计算板送显运算结果时
FF	40	一长串"咕噜"声		计算板送显结果结束时

二、重要工作单元

发声程序使用了 $F0～$FF 共 16 个零页单元和 11 个三页单元。零页单元的作用已在预备知识中予以介绍，现仅介绍三页单元的作用。

发声处理程序使用了三页中 $340～$34B 共 11 个单元。其中：

· $340 为音调随机数发生器，用于控制第三主声部在复位画面中发出"滴哒"声，取值范围为 H20～H80。

· $341 为音调升、降调控制器，由它控制 $340 产生由升调转为降调、再由降调转为升调的音调数据。当 $341=0 时，$340 为降调随机数发生器；当 $341=1 时，$340 为升调随机数发生器。

· $343 为第三主声部音调粗调控制器。

· $344～$34B 的作用见表 3－33。

表 3－33　三个主声部占用三页单元汇总表

发声声部	发声数据暂存单元	发声工作单元	发声计数器(音长控制)
第一主声部	344	347	34A
第二主声部	345	348	34B
第三主声部	346	349	

3·6·3　$FC 控制发声过程详析

一、复位画面中三种音响的发声过程

1、$FC=H70 的发声过程

我们知道，$FC=H70 时将发出连续 6 秒钟的"嘟嘟"声。下面我们讨论这一发声过程。

(1) 执行 $FC06 程序时，经调用发声处理子程序 $FC98、转 $FE14，因 $FC=H70、有

70 AND 10=10

而转 $FDC1。

(2) 执行 $FDC1，对第三主声部音量赋值，令 $4008=8；对第三主声部音调细调赋值，令 $400A=H7E。

(3) 转 $FDD5 对第三主声部发声音长控制器赋值，令 $F3=H0A。

(4) 因 $FC=H70，有

70 AND 40=40

故执行 $FDE1 时转 $FE0B，对第三主声部音调粗调单元赋值，令 $400B= $343=0($343 的初值为 0)。

(5) 通过(2)～(4)的三项操作，使第三主声部开始发声。

(6) 以后每中断一次、$F3 减一，只要 $F3≠0 就一直持续发声，发声 0.2 秒后，因 $F3=0 则对 $400A、$400B、$F3 重新赋值，发下一个"嘟"声。由于每次对 $400A、$400B 所置数值相同，所以发出持续的"嘟嘟"声。

2、$FC=H10 的发声过程

当"嘟嘟"声持续 6 秒钟后，$FC 由 H70 变为 H10。我们知道：当 $FC=H10 时将发出不规则的"滴哒"声。现在讨论其发声过程。

(1) 执行 $FE14，因

$FC AND 10=10

转 $FDC1，令 $4008=8、$400A=H7E。

(2) 转 $FDD5 对音长控制器赋值，令 $F3=H0A。因

$FC AND 40=0

所以取 $51 中的随机数送入 $400A，置音调细调值，并判断 $51 单元的随机数值与 HF0 相与的结果：

当 $51 AND F0=0 时，令 $343=1、$400B=1；

当＄51 AND F0≠0 时，令＄343＝0、＄400B＝0。
从而改变第三主声部的音调粗调值。

(3) 当经过五个中断周期(即发声 0·1 秒后)、＄F3＝5 时，将从音调随机数单元＄340 中取值送入＄400A、并令＄400B＝0，从而使后五个中断周期的发声变换音调。

(4) 当又发声 0.1 秒后、＄F3＝0，则又重新对＄F3 赋值，继续发声。可见＄F3 每次赋值发出两个音调，每个音调发声持续 0.1 秒钟。如此周而复始。由于＄51 单元数值的随机性和＄340 单元数值升、降调的周期性，从而使第三主声部音调细调值随机设定，发出不规则的"滴滴哒哒"的声响。

3、第二次＄FC＝H70 的发声过程

当不规则声响持续 12.8 秒后，又令＄FC＝H70。这时虽然又重复＄FC＝H70 的工作过程，但由于现在的＄343 取值在 0、1、2 之间随机确定，所以当第二次＄FC＝H70 时，往往发出高音调的连续的"嘟嘟"声。这就是两次＄FC＝H70 有时发声音调不同的原因。

二、CAL·板背景音乐的发声过程

在 CAL·板的工作中，有以进行速度的打击音乐伴随始终。它是由＄FC＝1 而启动的。该背景音乐共由三段乐章组成，下面分别讨论。

A、第一乐章

1、启动 CAL·板的背景音乐时＄FC＝1。当执行＄FE14 时，令 Y＝＄FC＝1。从而有

＄FC AND 3≠0

转＄FE2F，令＄FC 左移两次后送入 Y 寄存器，转＄FE49，以 Y 为变址指针，从＄FEE0 中取数送入＄F4、令＄F4＝0。

2、转＄FE55，以＄F4 为变址，令 Y＝＄F4，然后令＄F4＋1，从＄FEEC＋Y 中取出数据置入 A 寄存器、A＝71，通过＄FE5E～＄FE66 的一系列处理后，Y＝5。再从＄FFF4＋Y 中取出数据对第二主声部音长控制器＄F6 赋值、令＄F6＝H1E。

3、再令

A＝71 AND 3E＝30

调第二主声部赋值子程序，令

＄4005＝7F

＄4006＝AB

＄4007＝09

4、执行＄FE80，令＄4004＝94，从而令第二主声部发出"1"音。

5、在下一次中断期间，由于执行＄FE2F 时有

＄FC XOR ＄FB＝0

故转＄FE51 令音长计数器＄F6 减一，再转＄FE7A 令＄4004＝H93，使音量减小一级。

6、＄FE8B～＄FE92 为第二主声部八级音量数据，由它们控制第二主声部的发声音量逐渐衰减，最后关闭音量。第二主声部的发声响度曲线如图 3－8。

7、从图 3－8 可见，二声部的第一个音符发声时间为 8 个中断周期(即 0.16 秒)，而音长控制器＄F6 的取值为 H1E，即 30 个中断周期(0.6 秒)，发声时间约占音长的四分之一，若以 0.2 秒为一个节拍，则"1"音的简谱节奏应为‖ 1·0 0 0 ‖

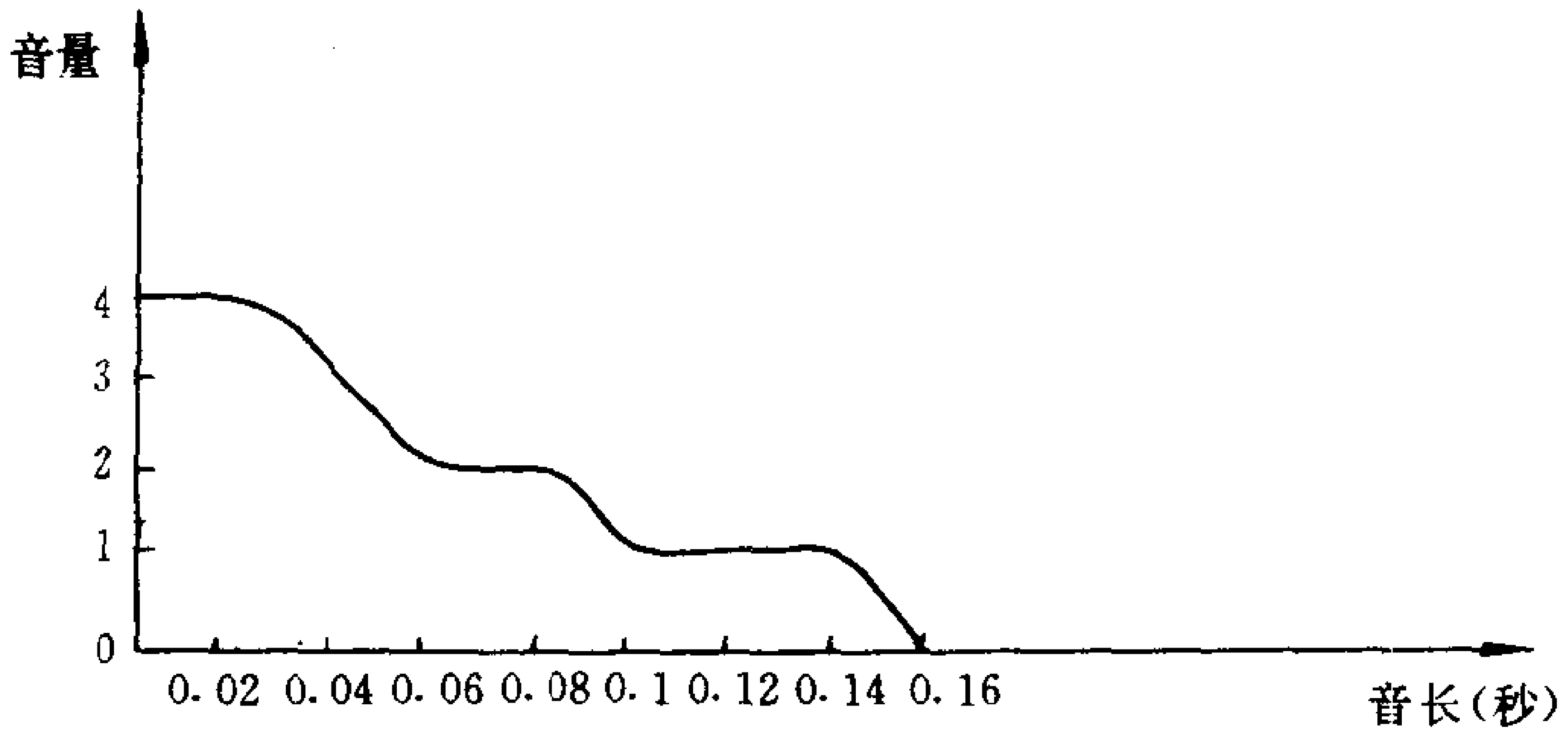


图 3－8　第二主声部发声音量衰减曲线

8、当经过 30 个中断周期、$F6＝0 后，则由 $F4 变址，从 $FEEC＋$F4 中取出 $F6 的赋值变址基数，A＝$FEED＝H6C，经过 $FE5E～$FE66 的运算处理，查出音长控制器 $F6 的值为 H0A，即 $F6＝H0A。

9、令　A＝6C AND 3E＝2C

调 $FC87，令 $4005＝7F

$4006＝DF

$4007＝09

发出"6＃"音。

10、"6＃"的发声时间仍为 0.16 秒，而 $F6 控制的延时时间为 0・2 秒，故"6＃"音的节奏近似为‖ 6＃ • 0 ‖。

表 3－34　CAL・板背景音乐第一乐章数据一览表

$F4	A 寄存器 ($FEEC＋$F4)	$F6	A AND 3E	$4006	$4007	对应的音　符	单符的节奏
0	71	1E	30	AB	09	1	1 • 0 0 0
1	6C	0A	2C	DF	09	6＃	6 ＃ • 0
2	AA	14	2A	FC	09	6	6 • 0 0
3	B0	14	30	AB	09	1	1 • 0 0
4	EC	28	2C	DF	09	6＃	6 ＃ • 0000
5	E6	28	26	3A	0A	5	5 • 0000
6	6A	0A	2A	FC	09	6	6 • 0
7	AA	14	2A	FC	09	6	6 • 0 0
8	66	0A	26	3A	0A	5	5 • 0
9	A2	14	22	80	0A	4	4 • 0 0
A	AA	14	2A	FC	09	6	6 • 0 0
B	E6	28	26	3A	0A	5	5 • 0 000
C	D8	28	18	57	0B	1	1 • 0 000
D	00						

以后各音符的发音过程基本相同，现将各有关数据列于表 3－34，以供参考。每个音符的

发音过程留给读者分析。

把表3—34中各音符按节奏排列就组出了CAL·板背景音乐的第一乐章的旋律。简谱如下：

‖1·0006#·0|6·001·00|

|6#·0 0 0 0|5·00 0 0|

|6·06·005·0|4·006·00|

|5·0 0 0 0|1·0 00 0‖

B、第二乐章

当$F4=H0D、执行$FE59时，A=$FEF9=0，则转$FE3B，把$51 AND 3的数值赋给$F4。假设现在$51中的随机数为H3E(取第206次循环数据)，则有$F4=2。以$FC左移两位后与$F4相或的值为变址指针，从$FED4数据区查出$F4的新值为$F4=$FEE2=H0E，转$FE55。这时将演奏CAL·板背景音乐的第二乐章对应音符数据见表3—35。

表3—35 CAL·板背景音乐第二乐章数据一览表

$F4	A寄存器 ($FEEC+$F4)	$F6	A AND 3E	$4006	$4007	对应的音符	单符的节奏
E	70	0A	30	AB	09	1	1·0
F	70	0A	30	AB	09	1	1·0
10	70	0A	30	AB	09	1	1·0
11	6C	0A	2C	DF	09	6#	6#·0
12	AA	14	2A	FC	09	6	6·00
13	B0	14	30	AB	09	1	1·00
14	AC	14	2C	DF	09	6#	6#·00
15	AA	14	2A	FC	09	6	6·00
16	E6	28	26	3A	0A	5	5·0000
17	6A	0A	2A	FC	09	6	6·0
18	6A	0A	2A	FC	09	6	6·0
19	6A	0A	2A	FC	09	6	6·0
1A	66	0A	26	3A	0A	5	5·0
1B	A2	14	22	80	0A	4	4·00
1C	AA	14	2A	FC	09	6	6·00
1D	67	1E	26	3A	0A	5	5·000
1E	58	0A	18	57	0B	1	1·0
1F	9C	14	1C	F9	0A	2	2·00
20	A0	14	20	A6	0A	3	3·00
21	00						

第二乐章的旋律为：

‖1·01·01·06#·0|6·001·00|

|6# ·0　0　6·0　0　|5·00　0　0|
|6·06·06·05·0　|4·006·00|
|5·0　0　0　1·0　|2·003·00‖

C、第三乐章

当 $F4=H21，在执行 $FE59 时，又有 A= $FF0D=0，从而第二乐章演奏结束。又取 $51 AND 3 的数值，假定现在 $51=H87（取第 301 次循环数据），则有 $F4=3，经移位换算后查出 $F4= $FEE3=H22，从而转入 CAL·板背景音乐的第三乐章。对应的音符数据见表 3－36。

表 3－36　CAL·板背景音乐第三乐章数据一览表

$F4	A 寄存器 ($FEEC+ $F4)	$F6	A AND 3E	$4006	$4007	对应的音　符	单符的节奏
22	DC	28	1C	F9	0A	2	2·0 000
23	A0	14	20	A6	0A	3	3·0 0
24	5C	0A	1C	F9	0A	2	2·0
25	60	0A	20	A6	0A	3	3·0
26	E2	28	22	80	0A	4	4·0 000
27	A6	14	26	3A	0A	5	5·0 0
28	62	0A	22	80	0A	4	4·0
29	66	0A	26	3A	0A	5	5·0
2A	EA	28	2A	FC	09	6	6·0 000
2B	AC	14	2C	DF	09	6#	6#·0 0
2C	6A	0A	2A	FC	09	6	6·0
2D	6C	0A	2C	DF	09	6#	6#·0
2E	F0	28	30	AB	09	1	1·0 000
2F	E2	28	22	80	0A	4	4·0 000
30	00						

第三乐章的旋律为：

‖2·0000|3·0　0　2·03·0|
|4·0000　|5·0　0　4·05·0|
|6·0000　|6# ·0 06·06# ·0|
|1·0000　|4·0　0　0　0　‖

通过以上讨论可见，由于 $51 取值的随机性，使三段乐章的组合也带有随机性，从而使背景音乐变化繁复、丰富多彩。这给我们以较少的数据演奏较复杂的音乐提供了极具参考价值的范例。

若把以上三段乐章随机组合，并考虑到人耳的听觉惰性，把每个音符的发声时间由 3/4 拍近似为一拍，然后再把每两拍合为一拍，则可组出一段很动听的乐章。示范如下：

‖：10 06#60 10 |6#00　5·00|
| 66 05 40　60　|50　0　10　0　|

10 06# 60 10	6#0 0 50 0
66 05 40 60	50 0 10 0
11 16# 60 10	6#0 60 50 0
66 65 40 60	50 01 20 30
20 0 30 23	40 0 50 45
60 0 6#0 66#	10 0 40 0
11 16# 60 10	6#0 60 50 0
66 65 40 60	50 01 20 30
20 0 30 23	40 0 50 45
60 0 6#0 66#	10 0 40 0
10 06# 60 10	6#0 0 50 0
66 05 40 60	50 0 10 0 : ‖

实际上,CAL·板的背景音乐也确实如此演奏。

三、MES·板背景音乐的发声过程

MES·板背景音乐的发声过程与CAL·板相同,不再赘述,仅给出背景音乐的旋律,读者可自己分析一遍。

MES·板背景音乐由$FC=2启动,音乐数据在$FF1D～$FF6E。它也由三段乐章组成,各乐章的旋律如下:

1、第一乐章旋律

‖ 1·0 0 6·0 0 | 4·0 0 5·0 6·0 |
| 6#·0 0 5·0 0 | 3·0 0 4·0 5·0 |
| 6·0 0 4·0 0 | 2·0 0 3·0 4·0 |
| 5·0 0 3·0 0 | 1·0 0 5·0 0 ‖

2、第二乐章旋律

‖ 1·0 6·0 1·0 6·0 | 1·0 6·0 1·0 6·0 |
| 6#·05·0 6#·05·0 | 6#·05·0 6#·05·0 |
| 6·0 4·0 6·0 4·0 | 6·0 4·0 6·0 4·0 |
| 5·0 1·0 2·0 3·0 | 4·0 5·0 6·06#·0 ‖

3、第三乐章旋律

‖ 1·06#·0 6·0 1·0 | 6#·06·0 5·06#·0 |
| 6·0 5·0 4·0 6·0 | 5·0 1·0 2·0 3·0 |
| 1·06#·0 6·0 1·0 | 6#·06·0 5·06#·0 |
| 6·0 5·0 4·0 6·0 | 5·0 ‖

四、第三种背景音乐介绍

若仔细分析$FEE0～$FF78的背景音乐数据区,我们可以发现数据区中还暗藏了一段背景音乐数据,我们可以很方便地调出,用在我们自己的游戏程序中。

令$FC=3,可以调出第三种背景音乐,它也由三段乐章组成,其旋律如下:

4·0 2·0 6#·02·0	4·0 2·0 6#·02·0
5·02#·06#·02#·0	5·02#·06#·02#·0
4·0 1·0 6·0 1·0	4·0 1·0 6·0 1·0

五、$FC=H68 时的发声过程

在对话板中，电脑等待操作者输入时发出的"哆哆"声是令$FC=68实现的。其发声过程如下：

1、执行$FE14时转$FDB6，令$FD加3。

2、置第三主声部音量数据，令$4008=8。

3、置第三主声部音调细调值，令$400A=H7E。

4、对音长控制器赋值，令$F3=7。

5、对第三主声部音调粗调单元赋值，令$400B=$343(等于0或1)，从而启动第三主声部发声。

6、在执行主控程序中，执行$FC33时，对第二副声部包络单元赋值，令$4011=$FD。由于$FD的初值为3、终值为H60。可使第三主声部发出的"哆哆"声由强到弱，最后维持轻微的"哆哆"声。(注——$4011可影响第三主声部发声。)

除对$FC进行以上赋值外，我们还可以使用$FC控制发出各种效果音。如：

$FC=5可发出"嗡嗡"声中伴有汽船声。

$FC=6可发出"呜呜"声中伴有滴滴声。

$FC=7可发出低沉的"嗡……"声中伴有"嗒嗒"声。

$FC=H50可发出无规则的"滴哒"声，音调呈波浪式变化等。这些留给读者分析。

3·6·4 $FE 单元控制发声过程详析

由表3-32可查出，在对话系统中由$FE单元控制发出的效果音有六种。下面分别讨论这些效果音的发声规律。

一、$FE=1 的发声过程分析

$FE=1是换板时发出的效果音。发声控制程序在$FE93～$FEDF。发声操作是：由$FA取$FE单元的发声数据启动、控制发声；由$F7控制发声音长；由$F5变址从数据区$FF8B～$FFF3中读出发声数据，以决定发出的效果音的种类。控制过程如下：

1、执行$FE93程序，令$FA=$FE=1。

2、令$F5=$FF8B=8，令$F7=1。

3、以$F5为变址指针，令A=$FF8B+$F5=HB3送入$F9单元，令$F9=HB3。

4、令$F5+1，以$F5为变址指针，令A=$FF8B+$F5=H30。

5、调$FC90第三主声部赋值子程序，对音调单元赋值：令$400A=HAB、$400B=9，置发声音调为"1"音。

6、取$F9单元的数据与H70进行"与"运算后，再右移两位置入$4008单元，以设置音色音量数据，有$4008=H0C。

7、取$F9单元的数值与H0F进行"与"运算后送入$F7，以控制发声音长，有$F7=3。

8、以后每次中断令$F7减一。当$F7=0时再取下一个数据，直到读数为0时，执行$FEBB，令$FA=0、$4008=0、关闭发声。

为便于分析，特把$FE=1时的发声数据及有关单元的对应关系列于表3-37。

表 3—37 $FE=1 时的效果音数据一览表

$F5	$F9	$FF8B+$F5	$400A	$400B	$4008	音长 $F7	音 符
8	B3						
9		30	AB	09	0C	03	1
A		38	52	08	0C	03	3
B		3E	1C	08	0C	03	5
C		48	D4	08	0C	03	1 *
D		50	A8	08	0C	03	3 *
E		56	8D	08	0C	03	5 *
F		60	69	08	0C	03	1 * *
10		00					

注——音符一栏中的一个“ * ”表示高一个八度音。

发声旋律为

‖1 3|5 1̇|3̇ 5̇|1̈ 0‖

由于每个音符的发声时间极短(仅有 0.06 秒),七个音符的发声时间仅有 0.42 秒,所以发出的效果音为一串嘟噜声。

二、$FE=2 的发声

当 COMPUTER 栏每次送显完毕时令 $FE=2,发出“叽哩咕噜”的声响。其发声过程同 $FE=1。下面仅把发声数据列于表 3—38。

表 3—38 $FE=2 时的效果音数据一览表

$F5	$F9	$FF8B+$F5	$400A	$400B	$4008	音长 $F7	音 符	音符的时值
11	92				04	02		
12		52	9F	08	04	02	4̇	4
13		60	69	08	04	02	1̈	1
14		50	A8	08	04	02	3̇	3
15		5E	70	08	04	02	7̇	7
16		4E	B2	08	04	02	2̇#	2#
17		5C	76	08	04	02	6̇#	6#
18		4C	BD	08	04	02	2̇	2
19		5A	7E	08	04	02	6̇	6
1A		4A	C8	08	04	02	1̇#	1#
1B		58	85	08	04	02	5̇#	5#
1C	A4				08	02		
1D		48	D4	08	08	04	1̇	1̇
1E		60	69	08	08	04	1̈	1̈
							⋮	⋮
1F		78	34	08	08	04	1	1̈

发声旋律为‖ 4̇1̈ 3̇7̇ | 2̇#6̇# 2̇6̇ | 1̇#5̇# 1̇ | 1̈ 1̈ ‖

由于每拍发声时间为 0.08 秒钟，所以 $FE=2 时的效果音发声时间为 0.64 秒。

三、$FE=4 的发声

$FE=4 所发出的效果音响用于从复位画面进入对话板时。发声数据见表 3－39。

表 3－39　$FE=4 时的效果音数据一览表

$F5	$F9	$FF8B+$F5	$400A	$400B	$4008	音长 $F7	音　符
21	A4				08	04	
22	A4	48	D4	08	08	04	1*
23	A4	56	8D	08	08	04	5*
24	A4	4C	BD	08	08	04	2*
25	A4	5A	7E	08	08	04	6*
26	A4	50	A8	08	08	04	3*
27	A4	5E	70	08	08	04	7*
28	A4	52	9F	08	08	04	4*
29	A4	60	69	08	08	04	1**
2A		00					

注——音符一栏中的一个"*"表示高一个八度音。

其旋律为：

‖ 1　5 | 2　6 | 3　7 | 4　1 ‖

发声时间为 0.64 秒。

四、$FE=H10 和 $FE=H40 的发声

$FE=H10 可控制发出操作者按回车键时的"嘟噜"声；$FE=H40 可控制发出按功能键或操作者输入字符时的"发啦"声。它们的发声数据如表 3－40 所示：

表 3－40　$FE=10、$FE=40 时的效果音数据一览表

$FE	$F5	$F9	$FF8B+$F5	$400A	$400B	$4008	音长 $F7	音　符
10	2B	A2				08	02	
	2C	A2	48	D4	08	08	02	1*
	2D	A2	50	A8	08	08	02	3*
	2E	A2	56	8D	08	08	02	5*
	2F	A2	60	69	08	08	02	1**
40	5C	A4				08	04	
	5D	A4	52	9F	08	08	04	4*
	5E	A4	5A	7E	08	08	04	6*

注——音符一栏中的一个"*"表示高一个八度音。

它们的发声旋律分别为

‖ 13　51 ‖ 和 ‖ 4　6 ‖

发声时间均为 0.16 秒。

五、$FE=20 时的发声

$FE=H20 时，第三主声部发出一段小号演奏的得胜音乐。其旋律为：

‖ 1 11 | 1 33 | 3 1 | 3 33 |

| 3 55 | 5 3 | 5 55 | 5 11 |

| 11— ‖

这一段音乐对话工作系统中没有使用。

3·6·5 ＄FF 单元控制发声过程详析

由＄FF 单元控制可令第一主声部发出 COMPUTER 栏送显每个字符时的“哒哒”声；OPERATOR 栏每送显一个字符时的“当当”声；CAL·板每送显一个运算结果字符时的“啾……”声；还有 CAL·板送显运算结果完毕后的一长串“咕噜”声。由＄FF 控制发声的处理程序在＄FD29～＄FDB5。其发声过程如下：

一、＄FF＝2 的发声过程

1、执行＄FD5B，把＄FF 单元的内容存入 Y 寄存器。

2、对＄FF 单元进行右移位操作。现在＄FF＝2，即＄FF 的 D1 位为 1，故转＄FD29。

3、执行＄FD29 令＄F0＝＄FF＝2。

4、对第一主声部音长控制器赋值，令＄F1＝3。

5、令 A＝H5A、X＝H80，调第一主声部赋值子程序＄FC6D，令

＄4000＝80	＄4001＝7F
＄4002＝7E	＄4003＝08

令第一主声部发“$\dot{6}$”音。

6、令＄F1 减一。

7、在下一次中断期间，因＄F0＝2 转＄FD4C，令＄F1 减一。

8、在第三次中断时转＄FD4C，令＄F1－1，有＄F1＝0，则令＄F0＝0、＄4000＝H90，使发音以变音色、快速减音量而结束，从而产生类似于敲击金属的声响。

二、＄FF＝4 的发声

1、执行＄FD5B，把＄FF 单元中的内容存入 Y 寄存器。

2、对＄FF 进行右移位。现在＄FF 的 D2 位为 1，所以转＄FD38。

3、执行＄FD38，令＄F0＝＄FF＝4。

4、令 A＝48，转＄FD2D，令＄F1＝3。

5、令 X＝80，调第一主声部赋值子程序＄FC6D，令

＄4000＝80	＄4001＝7F
＄4002＝D4	＄4003＝08

使第一主声部发“$\dot{1}$”音。

6、令＄F1 减一，转后续程序。

7、在第二次中断执行＄FD5D 时，因＄F0＝4，故转＄FD4C，令＄F1 减一。

8、在第三次中断时＄F1＝0，则令＄F0＝0、＄4000＝H90，使发音以变音色、快速减音量而结束。

三、＄FF＝8 的发声过程

1、执行＄FD5B，把＄FF 单元中的内容存入 Y 寄存器。

2、对＄FF 单元进行右移位，现在＄FF 的 D3 位为 1，所以转＄FD3E。

3、执行＄FD3E，令＄F0＝8。调音量赋值子程序＄FD11，以＄51 单元的随机数与 7 相“与”的值为变址指针存入 Y 寄存器。从＄FCE6～＄FD10 音色、音量数据区取数，令 X＝＄FD01＋Y、A＝＄FD09＋Y，以便对＄4000、＄4001 单元赋值。因为＄51 AND 7 的结果有 0

～7 八个取值，所以对应于 X、A 寄存器也有八组数据，现将其列于表 3－41。

表 3－41　$FF＝8 时的发声数据一览表

$51AND7	X 寄存器	Y 寄存器	$4000	$4001	$4002	$4003	音符
0	48	8C	80	8C	D4	08	$\dot{1}$
1	4C	8C	80	8C	BD	08	$\dot{2}$
2	4E	83 或 8C	80	83 8C	B2	08	$\dot{2}$#
3	50	83 或 8C	80	83 8C	A8	08	$\dot{3}$
4	54	83 或 8C	80	83 8C	96	08	$\dot{4}$#
5	58	8C	80	8C	85	08	$\dot{5}$#
6	5C	83	80	83	76	08	$\dot{6}$#
7	60	83	80	83	69	08	$\dot{1}$

假定 $51 AND 7＝1 则有 X＝H4C、A＝H8C，令 Y＝A＝H8C、A＝X＝H4C，返回 $FD40。

4、对音长控制器 $F1 赋值，令 $F1＝5。

5、令 X＝H80，调 $FC6F 子程序，令

$4000＝80　　　　$4001＝8C

$4002＝BD　　　　$4003＝08

令第一主声部发出“$\dot{2}$”音。

6、令 $F1 减一，执行后续程序。

7、在第二次中断，由 $F0＝8 转 $FD4C，令 $F1 减一。

8、经过五个中断周期，$F1＝0，令 $F0＝0、$4000＝H90 结束发声。

由于 $51 取值的随机性，所以 CAL・板送显运算结果字符所发出的“啾……”声的音调各不相同。

四、$FF＝H40 的发声过程

1、执行 $FD5B，把 $FF 单元的内容存入 Y 寄存器。

2、对 $FF 单元进行右移位操作，现在 $FF 的 D6 位等于 1，故转 $FD87。

3、执行 $FD87，令 $F0＝H40、$F1＝H3E。

4、令 A＝H18 调 $FC75 对第一主声部赋值，令 $4002＝H57、$4003＝H0B(发“$\underset{\cdot}{1}$”音)。

5、令 $4001＝H86，检查 $F1 的数值，令 $4000＝H90、令 $F1 减一。

6、当以 $4001＝H86，发声 0.14 秒后(7 个中断周期)，$F1＝H38，则令 $4001＝HBD、$4000＝H90，使“$\underset{\cdot}{1}$”音改变音调和包络，令 $F1 减一。

7、在随后的中断处理中($F1＝H37)，令 $4000＝H9A。

8、在下一次中断($F1＝H36)，又令 $4000＝H90。

9、令 7、8 的音量交替变换，直到发声 1.06 秒以后，则令音量($4000)从 H9A 开始每发音 0.02 秒，音量值减一。当音量值从 H9A 减到 H90 时则令 $F0＝0，结束发声。因此，$FF＝H40 可一次发声 1.24 秒。为便于仔细观察 $FF＝H40 时的发声过程，特将发声过程中 $4000、$4001 两单元内容的变化情况列于表 3－42。

表 3－42　$FF=40 发声过程中数据变化一览表

$F1	$4000	$4001	$F1	$4000	$4001
3E	90	86	2E	90	BD
3D	9A	86	2D	9A	BD
3C	90	86	2C	90	BD
3B	9A	86	2B	9A	BD
3A	90	86	2A	90	BD
39	9A	86	29	9A	BD
38	90	BD	28	90	BD
37	9A	BD	27	9A	BD
36	90	BD	26	90	BD
35	9A	BD	25	9A	BD
34	90	BD	24	90	BD
33	9A	BD	23	9A	BD
32	90	BD	22	90	BD
31	9A	BD	21	9A	BD
30	90	BD	20	90	BD
2F	9A	BD	1F	9A	BD
1E	90	86	0E	90	BD
1D	9A	86	0D	9A	BD
1C	90	86	0C	90	BD
1B	9A	86	0B	9A	BD
1A	90	86	0A	90	BD
19	9A	86	09	99	BD
18	90	BD	08	98	BD
17	9A	BD	07	97	BD
16	90	BD	06	96	BD
15	9A	BD	05	95	BD
14	90	BD	04	94	BD
13	9A	BD	03	93	BD
12	90	BD	02	92	BD
11	9A	BD	01	91	BD
10	90	BD	00	90	BD
0F	9A	BD			

从表 3－42 可以明显看出，$FF=40 的发声是交替关断、开通的（$4000=90 时关断，$4000=9A 时开通），所以它控制发出的声响是断续的。

3·6·6　发声工作过程小结

1、通过以上分析可见：三个主声部的发声互相独立，因此可以取任一个或两个主声部为背景音乐伴奏，用另一个发出效果音。

2、对话工作系统的三段背景音乐可通过对 $FC 单元置入不同的数值而调出，加入到我们自己的程序中。加入方法是：在系统中断程序中加入调用 $FC 控制发声的处理程序 $FE14。

3、我们还可以借鉴用 $FC 控制发声的处理程序、按 $FC 的数据格式编写我们所希望的背景音乐数据，演奏自己谱写的背景音乐。这也是音乐编程中最有效、最简单、最方便的方法。

为方便移植，特对$FC发声数据分析如下：

(1) 发声数据中的D5～D1位决定音调，它相对于$8006数据区的偏移量。其中D5、D4为高位、取值范围为0～3；D3～D1＋0为低位、取值范围为2～H0E中的偶数(2、4、6、8、A、C、E)。各音符与D5～D1位的对应关系见表3－43。

表3－43　用$FE14程序伴奏背景音乐的数据中D5～D1位与音符的对应关系

音　符	D5	D4	D3	D2	D1	音　符	D5	D4	D3	D2	D1
$\underset{:}{1}$	0	0	0	0	0	$\underset{\cdot}{3}$	1	0	0	0	0
$\underset{:}{1}\#$	0	0	0	0	1	$\underset{\cdot}{3}\#\underset{\cdot}{4}$	1	0	0	0	1
$\underset{:}{2}$	0	0	0	1	0	$\underset{\cdot}{4}\#$	1	0	0	1	0
$\underset{:}{2}\#$	0	0	0	1	1	$\underset{\cdot}{5}$	1	0	0	1	1
$\underset{:}{3}$	0	0	1	0	0	$\underset{\cdot}{5}\#$	1	0	1	0	0
$\underset{:}{3}\#\underset{:}{4}$	0	0	1	0	1	$\underset{\cdot}{6}$	1	0	1	0	1
$\underset{:}{4}\#$	0	0	1	1	0	$\underset{\cdot}{6}\#$	1	0	1	1	0
$\underset{:}{5}$	0	0	1	1	1	$\underset{\cdot}{7}$	1	0	1	1	1
$\underset{:}{5}\#$	0	1	0	0	0	$\underset{\cdot}{7}\#1$	1	1	0	0	0
$\underset{:}{6}$	0	1	0	0	1	1#	1	1	0	0	1
$\underset{:}{6}\#$	0	1	0	1	0	2	1	1	0	1	0
$\underset{:}{7}$	0	1	0	1	1	2#	1	1	0	1	1
$\underset{:}{7}\#\underset{\cdot}{1}$	0	1	1	0	0	3	1	1	1	0	0
$\underset{\cdot}{1}\#$	0	1	1	0	1	3#4	1	1	1	0	1
$\underset{\cdot}{2}$	0	1	1	1	0	4#	1	1	1	1	0
$\underset{\cdot}{2}\#$	0	1	1	1	1	5	1	1	1	1	1

(2) 发声数据中的D0、D7、D6三位决定发音长度的地址偏移量，取值范围为0～5(注——D0、D7、D6三位依次排列。)使用中根据它们的取值从音长数据区$FFF4～$FFF9中取出音长值。0～5对应的音长时值依次为0.16秒、0.2秒、0.4秒、0.8秒、1.6秒、0.6秒。它们的对应关系见表3－44。

表3－44　$FE14背景音乐程序中的数据、音长参数对照表

控制音长	音　长　(秒)	数据中的对应位取值		
		D0	D7	D6
08	0·16	0	0	0
0A	0·20	0	0	1
14	0·40	0	1	0
28	0·80	0	1	1
50	1·60	1	0	0
1E	0·60	1	0	1

(3) 我们还可以自己设定音长数据区，以演奏更复杂的音乐。

4、如果把

"JSR　$FD5B"

加入中断程序中，则不仅可以用＄FC控制发出各效果音和调出背景音乐，还可以用＄FE、＄FF控制发出各效果音。所不同的是：＄FC控制、一次赋值后可持续发声，而＄FE、＄FF控制则每赋值一次只发声一次，这是由于每次中断处理结束时对＄FE、＄FF置0的缘故。

5、＄FC06既是对话工作系统的发声处理程序也是电脑游戏机的发声示范程序。因此，它对我们在音响程序的设计中具有很好的借鉴作用。

6、我们还可以对＄FC置入H21、H22、H23、H25、H26、H27、H41、H42、H43、H45、H46、H47、H61、H62……、HE6、HE7等值，以发出不同的背景噪声，适应不同场合的需要。

3·7 音乐板工作过程分析

使用音乐板，可以在上、中、下三段音域任意输入音符进行谱曲练习、学习和声技巧、检验和声效果等。由于上、中、下三段的发音分别对应三个主声部，因此可以在谱曲操作中仔细体会三个声部之间的和声配合关系。更为奇妙的是：在音乐板还可以逆序演奏音乐。由于我们已经习惯了顺序演奏的音乐旋律，当逆序演奏时旋律突变，因而韵味别具、情调炯然，将把你带入另一音乐天地。

3·7·1 音乐板工作状态的建立

在询问“MUSICBOARD”

NEED?

时键入“OK”或在任意询问中键入“MUS·”回车后，将进行音乐板工作状态的建立操作：

1、令＄60＝C转＄D44A，清提示栏；令＄60＝6、＄C3＝4转＄D2DA，清两栏，送显“OK!”。

2、令＄60＝6、＄C3＝HC转＄D2DA，清两栏，送显

“MUSIC BOARD”

START

令＄FE＝2，发出COMPUTER栏送显完毕后的“叽哩咕噜”声。

3、令＄60＝9转＄CF06，关显示、清屏、清二页。

4、令＄60＝H12转＄D107，令＄62＝H10，调整＄60单元的赋值地址入口为＄CC2E。

5、令＄60＝8、＄6B＝9转＄CEF8，向各板画面控制器＄8A赋值，令＄8A＝9。

6、调＄EE43，送显“MUSICBOARD”字符标题及MUSIC板的各种文字说明。

7、令＄60＝8、＄6B＝6、对＄8A赋值，令＄8A＝6，调＄EE43，绘制MUSIC板工作画面的边框、横线等。MUSIC板的工作画面数据见图3－9、背景0页的配色单元数据分布见图3－10、MUSIC板的PPU配色区的配色见表3－45。

列次	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	行次
2000	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	1
2020	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	2
2040	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	3
2060	/	/	F3	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F7	/	/	4
2080	/	/	F4					EF	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	F0					F4	/	/	5
20A0	/	/	F4					EE			M	U	S	I	C		B	0	A	R	D				EE					F4	/	/	6
20C0	/	/	F4					F1	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	ED	F2					F4	/	/	7
20E0	/	/	F4																											F4	/	/	8
2100	/	/	F4																											F4	/	/	9
2120	/	/	F4																											F4	/	/	10
2140	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	11
2160	/	/	F4																											F4	/	/	12
2180	/	/	F4																											F4	/	/	13
21A0	/	/	F4																											F4	/	/	14
21C0	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	15
21E0	/	/	F4																											F4	/	/	16
2200	/	/	F4																											F4	/	/	17
2220	/	/	F4																											F4	/	/	18
2240	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	19
2260	/	/	F4																											F4	/	/	20
2280	/	/	F4																											F4	/	/	21
22A0	/	/	F4																											F4	/	/	22
22C0	/	/	F4		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		F4	/	/	23
22E0	/	/	F4						F	A	S	T	<	—	—	—	—	—	—	>	S	L	O	W						F4	/	/	24
2300	/	/	F4		F	1	—	F	7	:	c	d	e	f	g	a	b		F	8	:	P	L	A	Y		F9	FB		F4	/	/	25
2320	/	/	F4				(#)			:	C	D	E	F	G	A	B		$	:	R	E	P	E	A	T	FA	FC		F4	/	/	26
2340	/	/	F5	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F8	/	/	27
2360	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	28
2380	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	29
23A0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	30

注:HFF 用"/"代表
HB8 用"/"代表

图 3—9 MUSIC 板画面数据分布图

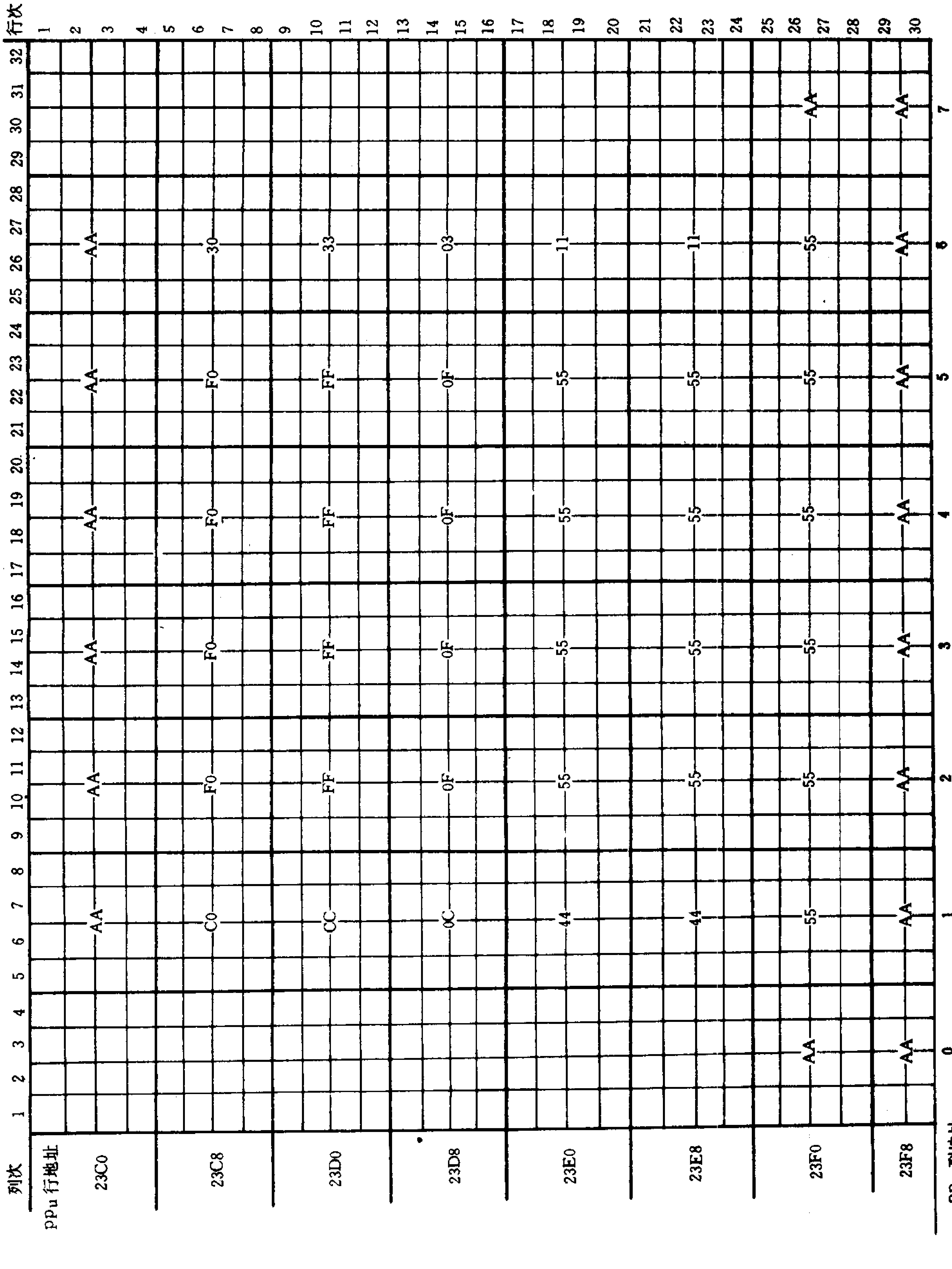


图 3－10 MUSIC 板配色分布图

表 3—45 MUSIC 板 PPU 配色单元数据一览表

PPU 地址	颜色代码	PPU 地址	颜色代码
3F00	0F	3F0C	0F
3F01	30	3F0D	30
3F02	15	3F0E	2C
3F03	05	3F0F	02
3F04	0F	3F10	0F
3F05	30	3F11	26
3F06	2C	3F12	30
3F07	11	3F13	15
3F08	0F	3F14	0F
3F09	30	3F15	29
3F0A	36	3F16	30
3F0B	07	3F17	1A

8、令＄60＝H25 转＄D139，进行以下操作：

(1) 令＄63D～＄7D3＝H20，清工作区。

(2) 从 MUSIC 数据库＄7200～＄731F 取回 MUSIC 板数据(第一次使用 MUSIC 板时全部为 H20)存入＄63D～＄7A4 单元。MUSIC 板工作单元与数据库单元的对应关系与其它各板不同，它采取分段分行存放的方式，其对应关系见表 3—46。

表 3—46 MUSIC 板工作单元与数据库单元的对应关系表

行次	列次	工作单元		数据库单元	
		起始地址	结束地址	起始地址	结束地址
第一行	上	63D	654	7200	7217
	中	655	66C	7260	7277
	下	66D	684	72C0	72D7
第二行	上	69D	6B4	7218	722F
	中	6B5	6CC	7278	728F
	下	6CD	6E4	72D8	72EF
第三行	上	6FD	714	7230	7247
	中	715	72C	7290	72A7
	下	72D	744	72F0	7307
第四行	上	75D	774	7248	725F
	中	775	78C	72A8	72BF
	下	78D	7A4	7308	731F

(3) 定义八个功能键 F1～F8 分别为

H63 H64 H65 H66 H67 H61 H62 HC8

(4) 定义演奏速度指示器的 SP 块

在 MUSIC 板画面的下部有演奏速度提示栏，提示栏中有六档速度选择，选定的速度由演奏速度指示器标出。该指示器由一个 SP 块担任，初始数据为：

＄220＝B7　　＄221＝DB　　＄222＝10　　＄223＝78

指示第三级演奏速度。

(5) 定义 6 个 SP 块组成三音程指示框，简称三音框。初始数据如下：

＄200＝37　　＄20C＝36

$201=D9	$20D=DA
$202=01	$20E=01
$203=1F	$20F=20
$204=3F	$210=3E
$205=D9	$211=DA
$206=01	$212=01
$207=1F	$213=20
$208=47	$214=46
$209=D9	$215=DA
$20A=01	$216=01
$20B=1F	$217=20

(6) 定义向上羽毛笔，占用 $2C0～$2DF 单元，数据同 CAL·板(从略)。

(7) 令 $A0～$DF=0，令 $FE=1 发出换板声。

(8) 令 $703A=1、$703B=1，置 MUSIC 板工作标志。

(9) 令 $60=26 转 $E423MUSIC 板工作程序，进入 MUSIC 板工作状态。至此，MUSIC 板工作状态建立过程结束。

3·7·2　音乐板工作程序简介

音乐板的主要工作程序位于 ROM 区 $E3DD～$E6E7。其中 $E3DD～$E422 为数据区、$E423～$E6E7 为程序区。另外，$F0B5～$F0D3 为辅助工作程序区、$FC06～$FCE2 为发声控制程序区。

音乐板的工作过程包括：

音符的输入
音符的连续演奏
音符的单步演奏
音符的逆序演奏
演奏速度的调整
屏幕编辑

六大过程。我们重点讨论音符的输入和演奏过程。

首先分段介绍 MUSIC 板的工作程序。

1、主控程序 $E423～$E438

主控程序完成对音乐板工作过程的控制。它根据按键的种类控制程序转移到相应的处理程序入口。基本分类如下：

- 按“Home”或“Home+Shift”键，控制转移到 $E449。
- 按“Home”键时转 $E455，把羽毛笔移到第一行上段首位置。
- 按“Home+Shift”键时转 $E469，进行 MUS·板的屏幕清除操作。
- 按“Esc”键时执行 $E439～$E445 进行退出 MUSIC 板的操作。
- 按音符键、功能键、系统操作键(Del、Ins、方向键、空格键等)时转 $E4F6 进行判断处理。

2、按键判断处理程序＄E4F6～＄E504

当按 Esc、Home、Home+Shift 以外的键时都由该段程序控制处理。它重点调用按键综合处理子程序＄EF2C 进行如下判断处理：

- 按方向键(→、←、↑、↓)则进行羽毛笔的移位操作。
- 按 Del 键则删去羽毛笔前面的字符。
- 按 Ins 键则在羽毛笔处插入空格。
- 按音符键(1～7、A～G、a～g、F1～F7)则进行输入音符的处理。
- 按空格键则进行演奏暂停处理。

3、演奏控制程序＄E505～＄E686

该段程序为 MUS·板的中心程序。由于它的程序量较大，控制功能也较多，因此它将是本节分析讨论的重点。该段程序主要完成以下控制：

(1) 控制每个音符数据的取入和送出。

(2) 控制音符的发声音长。

(3) 控制演奏速度。

(4) 完成一个音符演奏后令三音框移到下一个音符位置的控制。

(5) 单个音符的演奏控制。

(6) 逆序演奏的控制。

4、按"Home"或"Home+Shift"键的处理程序＄E449～＄E46C

按"Home"键的目的是把羽毛笔移到输入音符的首位，其操作过程同 CAL·板、MES·板基本相同。但按"Home+Shift "键时却与前两板有较大差别。音乐板清屏过程是：通过调用＄D197 子程序，首先由＄D1C7～＄D1D1 程序段控制连续清除三行后，跳过划横线的一行，再清除下面的三行。

5、重要子程序简介

(1) MUSIC 板退出操作子程序＄E439～＄E445。

(2) 把 MUSIC 板的内容存入数据库的子程序＄E46D～＄E4AF。

(3) 从数据库取回 MUSIC 板内容的子程序＄E4B0～＄E4F5。

(4) 定义三音框的子程序＄E687～＄E6D1。

(5) 定义演奏速度指示器的子程序＄E6D2～＄E6E7。

(6) 音符数据检索控制子程序＄F0B5～＄F0D3。

(7) 上段音符数据检索子程序＄E5CF。

(8) 中段音符数据检索子程序＄E5F4。

(9) 下段音符数据检索子程序＄E619。

3·7·3 音符的输入过程分析

输入 MUSIC 板的音符虽然也是字母或数字，但它的输入过程却与 CAL·、MES·板截然不同。在 MUSIC 板，每行音符分上、中、下三段，它们分别对应于第一、二、三三个主声部。同一行中不同段的音符，虽然外形相同但发音却有极大的差异。因此，每一段中音符位置的记忆单元＄2F、＄2E 已不象 CAL·板那样简单的累加，而是分段累加。因而当某行某段输入字符满行后的换行操作要比前两板复杂得多。为此，现将每一行各段音符的位置记忆单元的分配情况列于表 3—47 以供查用。

表 3—47 MUSIC 板各行各段音符位置单元分配表

行	上段(第一主声部)		中段(第二主声部)		下段(第三主声部)	
	$2F	$2E	$2F	$2E	$2F	$2E
第一行	00～17	0	18～2F	0	30～47	0
第二行	60～77	0	78～8F	0	90～A7	0
第三行	C0～D7	0	D8～EF	0	F0～07	0、1
第四行	20～37	1	38～4F	1	50～67	1

另外，每输入一个音符时，还要发出与音符对应频率的长衰减音，因此还要进行发声处理等操作。可见 MUSIC 板的音符输入过程较为复杂。为此，我们将把分析讨论重点放在与前面两板不同的部分。

一、上段音符的输入过程

以第一行上段输入"$\dot{1}$#"音为例。

我们知道，在上段输入"$\dot{1}$#"音时可以按"Shift+F1"键，也可以按大写字母"C"键，相应地$37=H43。屏幕上显示大写字母"C"，同时发出"$\dot{1}$#"的声调。下面讨论这一输入过程。

1、按"C"键后的初步处理

(1) 执行 MUSIC 板的工作程序$E423。因$37=H43 故转$E4F6。

(2) 令 X=4、调按键综合处理程序$EF2C 进行如下操作：

· 令$36=4、置 MUSIC 板工作标志。

· 因按"C"键则转$F03E，令 X=$2F=0，把键码 H43 送入$63D+X 单元，有$63D=H43。

· 由$F063 调$F0B5 音符数据检索控制子程序，再调用上段音符数据检索子程序$E5CF，从$E3DD～$E416 数据区中查出上段按"C"键对应的数据为 H4A，把 H4A 送入第一主声部发声数据暂存单元$344，令$344=H4A。

· 转$F06D、令$23=H10 以控制羽毛笔画圈。

· 返回主控程序$E4F8，按键的初步处理工作结束。

2、发声处理

(1) 在第二次中断期间(以按"C"键为第一次中断)，调用发声处理程序$FC06 时，经$FC0E～$FC1D 程序段的处理，把$344=H4A 的数值送入第一主声部发声工作单元$347中，令$347=H4A，同时令$344=0。

(2) 在第三次中断调用$FC06 时，通过调用$FC98，因$347=H4A，首先调第一主声部赋值子程序$FC69，令$4001=H7F、$4002=HC8、$4003=08；然后令第一主声部音长计数器$34A 复位(令$34A=0)；其次以$34A 右移一位的数值为变址从音色音量数据区$FCE6～$FD00 中取数对第一主声部的音色音量开关$4000 赋值，令$4000=H9F，第一主声部开始发声；最后令音长计数器$34A 加 1。当返回$FC0B 后，在执行$FC0E 时令$347复位，即$347=0。

(3) 在第四次中断调用$FC06 时，因$347=0，故转$FCA5，再根据$34A 右移一位的数值为变址取数赋于$4000。当前$34A=1、右移一位等于 0，所以仍有$4000=H9F。由此可见，音色音量数据区的每一个数据要发声 0.04 秒(两个中断周期)。

以后每两次中断改变一此$4000 单元的赋值。直到$34A=H36 时，其值不再增加，保持$34A=H36，恰好这时$4000=H90，第一主声部音量被关闭。因此，每输入一个音符要持续

发声H36个中断周期(约两秒钟),发声特点为从强到弱的衰减音。兹将音响变化数据列于表3—48供参考,并绘出音响衰减曲线于图3—11。

表3—48　MUS·板输入音符时的音量变化情况统计表

$34A	$4000	$34A	$4000	$34A	$4000
00	9F	13	96	26	92
01	9F	14	95	27	92
02	9E	15	95	28	91
03	9E	16	95	29	91
04	9D	17	95	2A	91
05	9D	18	94	2B	91
06	9C	19	94	2C	91
07	9C	1A	94	2D	91
08	9B	1B	94	2E	91
09	9B	1C	93	2F	91
0A	9A	1D	93	30	91
0B	9A	1E	93	31	91
0C	99	1F	93	32	91
0D	99	20	92	33	91
0E	98	21	92	34	90
0F	98	22	92	35	90
10	97	23	92	36	90
11	97	24	92		
12	96	25	92		

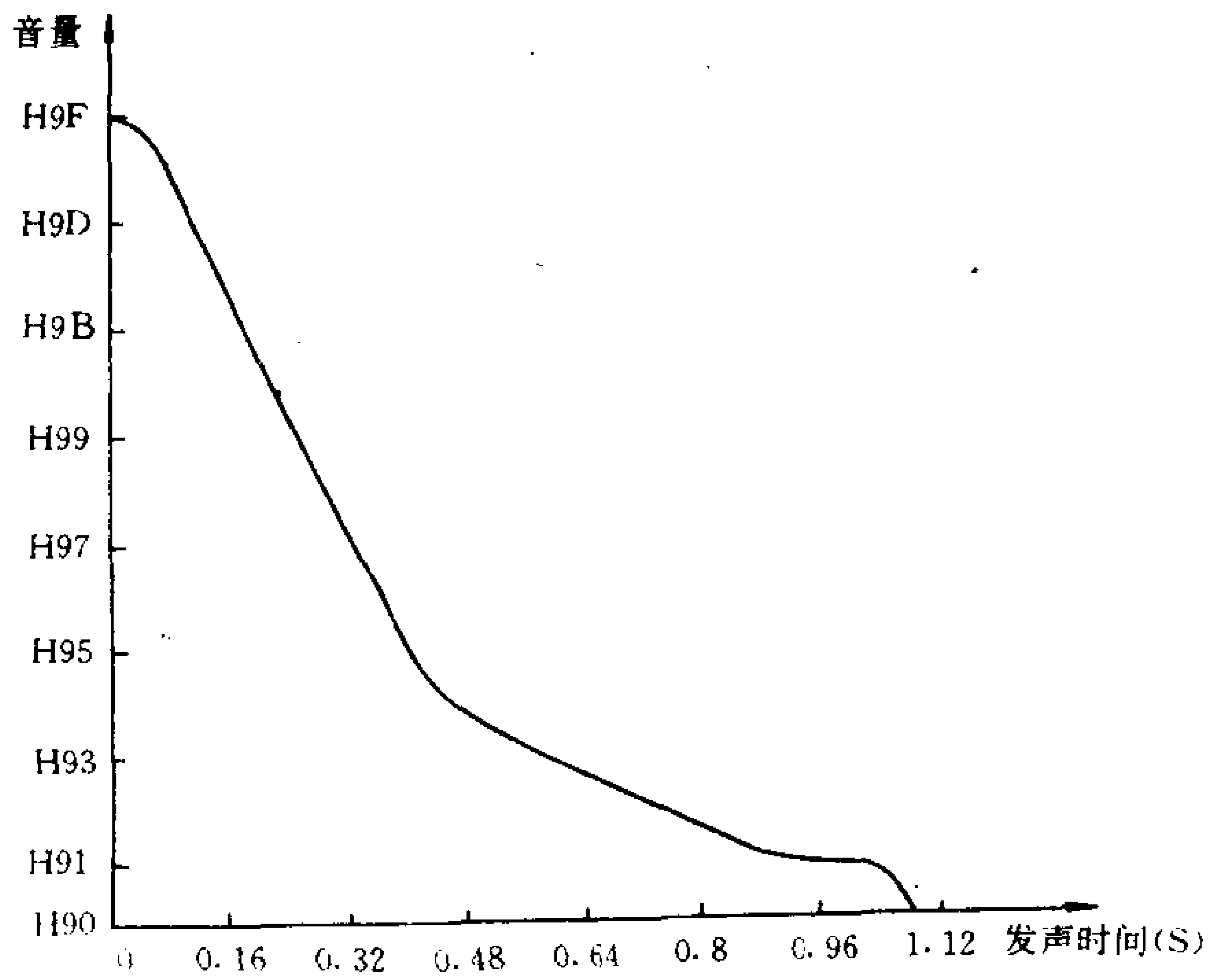


图3—11　第一主声部演奏音响衰减曲线图

3、羽毛笔的控制

羽毛笔的画圈控制过程与CAL·板完全相同。但画圈以后的处理却有较大的差别。当$23=1时,由$EF38转$F076,在调用$F115子程序时要进行输入是否满行的检查处理。若本行已输入了24个音符时则要进行换行处理。MUSIC板的换行操作要比CAL·板多调用3次$F1DE、$F259两个子程序,以使$2F连加3个H18、$31连加3个H08、$34连加3个H20,并令$2E紧跟$2F之后、$32紧跟$34之后连加3次进位“C”。如第一行上段输入满行时,有

$2F=18　　　　$2E=0

$ 30=DA　　　　　$ 31=3C

$ 32=20　　　　　$ 34=FB

连续调 3 次 $ F1DE、$ F259 后有

$ 2F=60(对应于表 3－47 中第二行上段之首)

$ 30=22(使羽毛笔的 X 坐标位于第二行之首)

$ 31=54(使羽毛笔的 Y 坐标位于第二行上段)

$ 32=21

$ 34=64(使 $ 32、$ 34 指向 PPU 的地址为 $ 2164)

4、上段音符按键对应的音符数据

从以上讨论可见，音符的按键码并非直接用于对一声部 $ 4002、$ 4003 单元赋值，而是每一音符键码对应一确定的音符数据，它就是音调数据区 $ 8006 的地址偏移量。为方便使用查阅，特把上段音符按键对应的音符数据汇集于表 3－49。

表 3－49　MUSIC 板上段音符按键数据一览表

按　键	键　码	音　符	数　据	4002	4003	备　注
c(F1)	63	1̇	48	D8	08	
d(F2)	64	2̇	4C	BD	08	
e(F3)	65	3̇	50	A8	08	
f(F4)	66	4̇	52	9F	08	
g(F5)	67	5̇	56	8D	08	
a(F6)	61	6̇	5A	7E	08	
b(F7)	62	7̇	5E	70	08	
C	43	1̇#	4A	C8	08	也可按 Shift+F1
D	44	2̇#	4E	B2	08	也可按 Shift+F2
E	45	3̇#	52	9F	08	也可按 Shift+F3
F	46	4̇#	54	96	08	也可按 Shift+F4
G	47	5̇#	58	85	08	也可按 Shift+F5
A	41	6̇#	5C	76	08	也可按 Shift+F6
B	42	7̇#(1̇)	60	69	08	也可按 Shift+F7
1	31	1	30	AB	09	
2	32	2	34	7C	09	
3	33	3	38	52	09	
4	34	4	3A	3F	09	
5	35	5	3E	1C	09	
6	36	6	42	FD	08	
7	37	7	46	E1	08	

二、中段音符的输入

中段音符的输入过程与上段基本相同,现仅将与上段不同的部分列出:

1、在＄F0B5 音符数据检索子程序中,要调用＄E5F4 中段音符数据检索子程序,查出的数据要送入第二主声部数据暂存单元＄345。

2、在发声处理中,通过＄FC0E～＄FC1D 程序段的处理,把＄345 的内容送入第二主声部工作单元＄348 中。经＄FCBC 调第二主声部赋值子程序分别对＄4005、＄4006、＄4007 单元赋值;由＄34B 控制对＄4004 赋值,控制第二主声部发声。发声衰减曲线同图 3－11。

3、音符按键与数据的对应关系与上段相比有所改变,兹汇集列于表 3－50。

表 3－50 MUSIC 板中段音符按键数据一览表

按 键	键 码	音 符	数 据	4006	4007	备 注
c(F1)	63	1	30	AB	09	
d(F2)	64	2	34	7C	09	
e(F3)	65	3	38	52	09	
f(F4)	66	4	3A	3F	09	
g(F5)	67	5	3E	1C	09	
a(F6)	61	6	42	FD	08	
b(F7)	62	7	46	E1	08	
C	43	1#	32	93	09	可按 Shift＋F1
D	44	2#	36	67	09	可按 Shift＋F2
E	45	3#	3A	3F	09	可按 Shift＋F3
F	46	4#	3C	2D	09	可按 Shift＋F4
G	47	5#	40	0C	09	可按 Shift＋F5
A	41	6#	44	EE	08	可按 Shift＋F6
B	42	7#	48	D4	08	可按 Shift＋F7
1	31	$\dot{1}$	48	D4	08	
2	32	$\dot{2}$	4C	BD	08	
3	33	$\dot{3}$	50	A8	08	
4	34	$\dot{4}$	52	9F	08	
5	35	$\dot{5}$	56	8D	08	
6	36	$\dot{6}$	5A	7E	08	
7	37	$\dot{7}$	5E	70	08	

三、下段音符的输入

下段音符的输入过程与上段基本相同,仅将与上段不同的部分列出如下:

1、在＄F0B5 音符数据检索子程序中要调用＄E619 下段音符数据检索子程序,查出的数据送入第三主声部数据暂存单元＄346。

2、在发声处理中,通过＄FC0E～＄FC1D 程序段的处理,把＄346 中的内容送入第三主声部工作单元＄349 中。通过调用三声部赋值子程序＄FC90 对＄400A、＄400B 赋以音调值;然后对＄4008 赋值,令＄4008＝H38,控制第三主声部发声。

3、下段音符按键与数据的对应关系见表 3－51。

表 3－51 MUSIC 板下段音符按键数据一览表

按 键	键 码	音 符	数 据	400A	400B	备 注
c(F1)	63	$\underset{\cdot}{1}$	30	AB	09	

按键	键码	音符	数据	400A	400B	备注
d(F2)	64	$\underset{\cdot}{2}$	34	7C	09	
e(F3)	65	$\underset{\cdot}{3}$	38	52	09	
f(F4)	66	$\underset{\cdot}{4}$	3A	3F	09	
g(F5)	67	$\underset{\cdot}{5}$	3E	1C	09	
a(F6)	61	$\underset{\cdot}{6}$	42	FD	08	
b(F7)	62	$\underset{\cdot}{7}$	46	E1	08	
C	43	$\underset{\cdot}{1}$#	32	93	09	也可按 Shift+F1
D	44	$\underset{\cdot}{2}$#	36	67	09	也可按 Shift+F2
E	45	$\underset{\cdot}{3}$#	3A	3F	09	也可按 Shift+F3
F	46	$\underset{\cdot}{4}$#	3C	2D	09	也可按 Shift+F4
G	47	$\underset{\cdot}{5}$#	40	0C	09	也可按 Shift+F5
A	41	$\underset{\cdot}{6}$#	44	EE	08	也可按 Shift+F6
B	42	$\underset{\cdot}{7}$#	48	D4	08	也可按 Shift+F7
1	31	1	18	57	0B	
2	32	2	1C	F9	0A	
3	33	3	20	A6	0A	
4	34	4	22	80	0A	
5	35	5	26	3A	0A	
6	36	6	2A	FC	09	
7	37	7	2E	C4	09	

3·7·4 MUSIC 板的演奏过程

MUSIC 板有多种演奏方式,如图 3—12 所示:

MUSIC 板演奏
- 1、连续演奏(按 F8 键后)
- 2、反复演奏(读到“$”后)
- 3、变速演奏 4、单音演奏(按空格键后)
  - 顺序单音演奏(按→键)
  - 逆序单音演奏(按←键)

图 3—12 MUSIC 板演奏方式示意图

演奏过程是 MUSIC 板的重要工作过程,下面对各演奏方式进行分别讨论。

一、连续演奏

音符输入过程结束,欲进行旋律演奏时可按 F8 键。按 F8 键时,$37=C8,将开始连续演奏。连续演奏的控制处理过程共分四个阶段:

1、读数阶段

(1) 由$EF2C 按键综合处理程序令$38=$39=$40=HC8。

(2) 因$38=HC8,故执行$E505 以后的演奏处理程序。

(3) 调$E46D 子程序,把 MUSIC 板的内容送入$7200~$731F 保存。各段数据的保存

地址对应于表 3－46。

(4) 令 $ 8B＝1、置演奏开始标志。调 $ EFE8 子程序定义向下羽毛笔，使羽毛笔插入画面右下方的墨水瓶中。

(5) 对音符变址指针 $ A0 置 0，调 $ E596 子程序，令 Y＝ $ A0，分别从 $ 7200＋Y 中取出上段音符的 ASCII 码，从表 3－49 中查出对应的数据送入 $ 344 单元；从 $ 7260＋Y 中取出中段音符的 ASCII 码，由表 3－50 查出对应的数据送入 $ 345 单元；从 $ 72C0＋Y 中取出下段音符的 ASCII 码，由表 3－51 中查出对应的数据送入 $ 346 单元。

(6) 根据 $ 8C 的取值（$ 8C 的初值为 2）从速度数据区 $ E41D～ $ E422 中取值赋给 $ 24，令 $ 24＝H10 以控制音符的发声时间。初值设定为每个音符发声 0.3 秒。

2、赋值阶段

当 $ 344～ $ 346 赋值后，在调用 $ FC06 发声子程序时，对三个主声部的工作单元 $ 347～ $ 349 赋值，控制三个主声部发声。发声时间由 $ 24 单元控制。

3、演奏中处理阶段

在以后执行主控程序 $ E423 的过程中，因 $ 8B≠0，故转 $ E52B 进行以下几方面的操作：

(1) 检查 $ A1（按空格键标志。按空格键时 $ A1≠0），若 $ A1≠0 则转 $ E55B 进行演奏暂停处理。

(2) 检查是否按方向键，若按方向键"→"或"←"则进行演奏速度的调整。

(3) 检查是否松开演奏键 F8，若没有松开则返回读数阶段。

(4) 若已松开 F8 键，且没有按空格键则转 $ E581，检查音长计数器 $ 24 单元。

若 $ 24≠0 则直接返回、持续发音。

若 $ 24＝0 则对 $ 24 重新赋值，调 $ E641 子程序、令 $ A0＋1，把三音框调整到下一个音框位置显示。

(5) 检查反复演奏标志 $ A4，若 $ A4≠0 则转 $ E511 进行读数阶段处理（读到" $ "时，令 $ A4≠0）；若 $ A4＝0 则转 $ E596 取下组音符，查出对应的数据送入 $ 344、$ 345、$ 346 单元。

(6) 反复重复以上？～" 的处理过程，直到 $ A0＝H60、即演奏完 MUSIC 板全部音符、三音框位于第四行尾时，由 $ E59B 控制转 $ E5BA 进行演奏结束处理。

4、演奏结束处理阶段

(1) 令 $ 8B＝0，置演奏结束标志。

(2) 转 $ E518，令 $ A0＝ $ A3＝ $ A4＝0。定义三音框在第一行首位显示。结束演奏过程。

二、反复演奏

反复演奏由反复演奏符" $ "启动。当 MUSIC 板在演奏中读到" $ "号时将进行如下操作：

1、转 $ E5C1，令 $ A4＝1（设置反复演奏标志）。

2、在下一次中断执行 $ E423 时，由 $ E593 转 $ E511 令 $ A0＝ $ A4＝0，从第一行首位开始重复连续演奏的第 1、2、3 项处理。

3、只要不发出暂停指令（按空格键），反复演奏就一直进行下去。

三、变速演奏

在连续或反复演奏中，若按左、右方向键可进行变速演奏。按左移键加快节奏、速度指示器向左移；按右移键减慢节奏、速度指示器向右移。共有六档演奏速度供选择使用，它们由 $ 8C

取值 0～5 指定，通过＄24 单元控制中断次数来控制发声音长。对应关系见表 3－52。

表 3－52　MUSIC 板演奏节奏与音长对照表

＄8C	＄24	音长(秒)	速度指示器位置
0	06	0.1	左起第一位
1	0B	0.2	左起第二位
2	10	0.3	左起第三位
3	15	0.4	左起第四位
4	1A	0.5	左起第五位
5	1F	0.6	左起第六位

1、减慢节奏

减慢节奏的处理程序在＄E52F～＄E53E 和＄E548～＄E54C。处理过程如下：

(1) 当按右移键时，＄40＝H1C，因而执行＄E535 以后的程序。

(2) 检查＄8C 是否已达到最低速度＄8C＝5，若＄8C＝5 则转＄E581 程序，不再调整演奏速度；若＄8C<5 则令＄8C 加一后转＄E548。

(3) 调＄E6D2 根据＄8C 的取值，调整演奏速度指示器的 X 坐标，令其在新的位置显示。

(4) 转＄E581 对＄24 置入新的速度值，使音符在＄24 的控制下改变发声音长。

2、加快节奏

加快节奏的处理程序在＄E53F～＄E54C，处理过程如下：

(1) 检查＄8C 是否已达到最快节奏：若＄8C＝0 则不再调整；若＄8C>0 则令＄8C 减一。

(2) 重复减慢演奏节奏的(3)、(4)操作。

四、单音演奏

进入单音演奏状态，首先要按空格键进行演奏暂停。按空格键后＄37＝H20、＄40＝H20，由＄E558 控制令＄A1＝1，置单音演奏标志，进入单音演奏状态。

1、顺序单音演奏

顺序单音演奏的处理程序在＄E56A～＄E575。顺序单音演奏时按右移键，这时三音框将随着右移键的按动顺序右移、顺序演奏音符，其操作过程如下：

(1) 检查是否按演奏键，若＄40＝HC8 则退出单音演奏状态。

(2) 若＄40＝H1C 则为顺序单音演奏，由＄E566～＄E575 程序控制进行。

(3) 检查＄A0 是否为最后一个音符：若＄A0＝H5F 则结束演奏(＄A0＝5F 是第四行尾标志)；若＄A0<H5F 则执行＄E570。

(4) 调＄E641，令＄A0 加一、令三音框的 X 坐标单元加 8。若三音框移到右边界(＄3E＝HE0)则令＄3E＝H20、＄3F＋H20，使三音框在下一行首显示；若＄3E<HE0 则转＄E663，令三音框在下一音符位置显示。

(5) 转＄E596 以＄A0 变址从＄7200 以后的单元中取出音符的 ASCII 码转化为发声数据存入＄344～＄346 单元，以启动发声。

(6) 一个音符发声完毕，若不按方向键则处于等待状态(暂停演奏状态)，直到按方向键时再演奏下一音符。

2、逆序演奏

逆序演奏的处理程序在＄E57A～＄E57F。处理过程如下：

(1) 调＄E667 检查＄A0 的取值：若＄A0＝0，表示三音框处于第一行首，则不再执行逆序单音演奏；若＄A0≠0 则令＄A0 减一、令＄3E 减 8。

(2) 检查＄3E 单元的内容：若＄3E＝H18 表示三音框已位于本行的最左端，则令＄3E＝HD8、＄3F－H20、使三音框移到上一行最右端显示；若＄3E>H18 则令三音框前移一个音符位置显示。

(3) 转＄E596 以＄A0 单元变址，从＄7200 以后单元中取出音符转化为发声数据送入＄344～＄346 单元启动发声。

(4) 一个音符发声完毕，若不继续按左移方向键则处于等待状态，直到按左移键时再演奏前一个音符。

3·7·5 MUSIC 板的退出操作

当 MUC·板工作完毕，欲退出该板时可按"Esc"键，这时＄37＝H1B，执行＄E439 的 MUS·板退出处理程序。

1、调＄F4B1 定义功能键子程序，定义 F1～F8 八个功能键为 HC1～HC8。

2、调＄E46D 子程序，把＄63D～＄7A4 单元的数据存入 MUSIC 板数据库＄7200～＄731F。

3、转＄DC9B 令＄703A＝H5A、＄703B＝HCC，置工作板结束标志。令＄FE＝1 发出换板声响。令＄60＝0、关闭 MUSIC 板工作程序入口。转入对话板画面的绘制，返回对话工作状态。

MUSIC 板工作分析小结

1、MUSIC 板的音符输入为分行分段存放方式。它使用＄42 单元作为上、中、下三段的工作标志。输入音符计数器＄2F、＄2E 中的数值、音符暂存单元＄63D～7A4、MUSIC 板数据库＄7200～＄731F 等单元都是分段使用的。其中＄63D～＄7A4 在一行中连续使用、行与行之间分段使用。即第一行上、中、下三段使用＄63D～＄684 连续的 72 个单元、第二行上、中、下三段使用＄69D～＄6E4 连续的 72 个单元、第三行上、中、下三段使用＄6FD～＄744 连续的 72 个单元、第四行上、中、下三段使用＄75D～＄7A4 连续的 72 个单元。而＄7200～＄731F 则是在一段中连续使用，段之间分区使用。即上段的四行使用＄7200～＄725F 连续的 96 个单元、中段的四行使用＄7260～＄72BF 连续的 96 个单元、下段的四行使用＄72C0～＄731F 连续的 96 个单元。其对应关系见表 3－46、表 3－47。

2、MUSIC 板音符的演奏工作过程：由＄E423 入口程序读出音符数据存入＄344～＄346 单元，由＄FC06 程序控制发声。通过演奏过程的分析讨论，重点加深领会＄FC06 发声程序中三个主声部的赋值和发声控制方法。

3、三个主声部的发声规律是不同的。第一、第二主声部具有相同工作性质，而第三主声部则要比前两个声部低一个八度音。即当对三个主声部的工作单元置入相同的数据时，第三主声部的发声音调要比前两个低八度。

4、MUSIC 板的最大发声音域为四个八度音，从"1～1"。前三个八度音还能演奏半音。因此，MUSIC 板基本能够满足一般 』前三个八度音还能演奏半音。因此，MUSIC 板基本能够满足一般音乐旋律的演奏需要。

第四章　F BASIC 工作系统简介

BS. 2A 版本是 F BASIC 的基础版本，后来的各种版本都是由 BS. 2A 版本扩展而成的。BS. 2A 版本共有 73 个保留字(其中有两个未使用)、组成 79 条指令。指令的解释程序占用了系统 ROM 区的 $80D7～$C127 约 16K 字节。用户程序存储区位于 RAM 区的 $703E～$7FFF 共 4030 个字节。F BASIC 的程序输入和屏幕显示使用背景零页，BG GRAPHIC 绘图则使用背景一页。F BASIC 和 BG GRAPHIC 工作状态的转换在选项画面中选定。在 F BASIC 状态下键入"SYSTEM"命令、在 BG GRAPHIC 状态按"Esc+Stop"键都可进入选项画面。因此，工作状态间的转换操作简单方便。

本章将通过对 F BASIC 各条指令的讨论，了解它们执行过程中的操作实质，查出可独立调用的系统子程序供编程中调用，达到以较小的程序量编写出较强功能程序的目的。

4·1　F BASIC 工作系统的基本特征

4·1·1　F BASIC 工作状态的建立

在对话工作系统的对话板询问"F BASIC NEED?"时，按"F1"键或键入"OK"，回车后将进行如下操作：

1、令 $60=C、$6B=0，清提示栏后令 $60=0。

2、执行 $CE26 令 $60=6、$6B=0、$C3=4，清两栏，送显"OK!"，令 $60=0，显示延时。

3、送显"F BASIC START"，延时结束后令 $60=0。

4、令 $60=9、$6B=0、转 $CF06，令 $2001=0，关显示、清屏、清二页、令 $60=0。

5、令 $60=1、$6B=0，在执行 $C44A 复位程序时转 $8000，再转 $80AD。

6、执行 $80AD，令 $ED=H4C、$EE=H0F、$EF=H88，设置新的中断程序入口 $880F。

7、调 $AA82 子程序，对零页单元置入初值：

(1) 令 $E0=HFD、$E1=H5F、$61=HE0，设定光标形状；

(2) 令 $4F=H20，置背景零页为显示页，调清屏子程序 $AB75 清背景零页；

(3) 调 $AB09 子程序，把背景配色代码的每组四个颜色代码从 $C3BF 数据区取出，分四次存入 $701～$704 单元，令 A=6 触发 $63 单元，把配色代码及数据写入 PPU 的背景配色区 $3F00～$3F0F；再把卡通配色代码的每组四个颜色代码取出存入 $701～$704 单元，令 A=7 触发 $63 单元，把卡通配色代码及配色数据置入 PPU 的卡通配色区 $3F10～$3F1F；

(4) 清功能键定义区 $480～$4FF，然后由 $AAD4～$AAF2 程序处理，从 $AB1D 数据区读出八个功能键的定义内容送入各自的定义单元，把各功能键定义为：

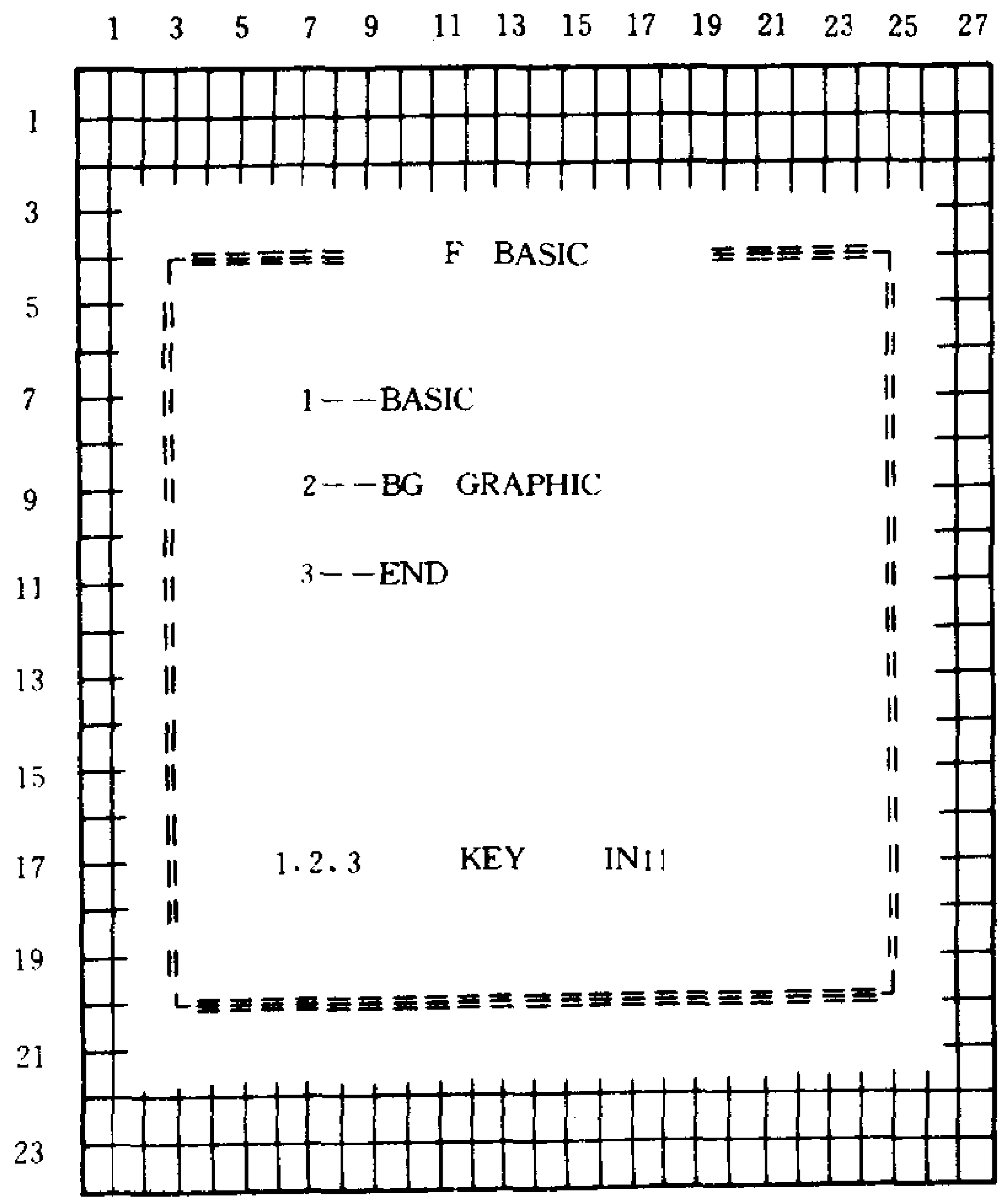
F1:LOAD(M)　　　　F2:PRINT

F3:GOTO　　　　F4:CHR$(

F5:SPRITE　　　　F6:CONT(M)

F7:LIST(M)　　　　F8:RUN(M)

8、调$8131子程序进行F BASIC选项画面的绘制。由$44单元控制画出F BASIC的选项画面如下图：



F BASIC选项画面

（调 $8A9C子程序向PPU置入数据画框；调$8154子程序显示选项字符）

9、调$ADA9显示闪烁光标；调$AF88读键子程序读取操作者选项。这时由$80C5～$80D0程序段封锁了除"1、2、3"三个数字键和STOP键外的所有按键，等待操作者选项。

若按"1"键则进入F BASIC工作状态；

若按"2"键则进入BG GRAPHIC工作状态；

弱按"3"键则返回复位画面。

10、按"1"键后，转$80F0进行进入F BASIC编程等待状态的处理：

(1) 检查$00单元：若$00=H59则转$8121。现在是第一次进入F BASIC，$00≠H59故顺序执行；

(2) 检查$703A单元：若$703A≠H5A则转$8127。由于从对话板退出时$703A=H5A、$703B=HCC，故顺序执行。

(3) 检查$703B单元：若$703B≠H33则转$8127。现在$703B=HCC故转$8127。调$814B，若$9E=H81则直接返回；若$9E≠H81则清背景一页后返回。

(4) 令$00=H59、转$81FF进行如下操作：

令S=HFF、设置堆栈顶；

调$852E、对系统零页有关单元置初值，清系统二页，令$200～$2FF=HEF；

调$8388、令$703E=$703F=$7040=0，从用户程序区首址开始连续置入三个零，表示程序区为空区；令用户程序区首址记录单元$05、$06指向$703E；程序区末址记录单元$07、$08指向$703F；变量区首址记录单元$1D、$1E指向$7040；变量区末址记录单元(自由RAM区首址单元)$1F、$20置入$7041；

调$AA09、显示进入F BASIC后的提示

"F BASIC BS. 2A

@ CNEDC SHEN ZHEN CO.

4030 BYTES FREE"

(5) 令$24=0、置CONT指令无效，令$703A=$703B=0，送显"OK"，进入操作者输入程序等待状态。

至此，F BASIC工作状态建立完毕。

4·1·2 F BASIC 工作系统的 RAM 分配

F BASIC 工作系统把 RAM 区进行了仔细的、确切的分工。特别是对＄0000～＄07FF 这 2K 字节分工明确、不许混用，否则将影响系统的正常工作，轻则造成运算结果错误；重则造成死机。只有全面掌握 RAM 中哪些单元禁止使用、哪些单元可以有条件使用、哪些单元可自由使用，才能保证我们编写的机器语言程序既能顺利运行又不影响系统的工作。我们把禁用单元前面加"＊＊"或"＊"号，有条件使用单元无"＊"号，自由使用的单元则不列出(有"＊＊"号的为特殊单元)。

1·系统零页分工

＊ ＄00:进入 F BASIC 工作系统标志

＄02:PALETB 指令中的计数器

＊ ＄03、＄04:RAM 自由区末址

＊ ＄05、＄06:用户程序区首址

＊ ＄07、＄08:用户程序区末址

＊ ＄09、＄0A:执行程序的变址指针

＊ ＄0B、＄0C:当前执行程序行的行号

＊ ＄0D、＄0E:下一行程序在内存中的首址

＊ ＄0F、＄10:翻译程序时暂存程序行的行号

＄11、＄12:STOP 中暂存当前执行程序行的行号

＄13、＄14:STOP 中暂存下一程序行首址

＄15、＄16:STOP 中暂存当前执行指令的地址

＊ ＄17:程序转向前的栈指针

＊ ＄18:读键工作单元。按 Ctrl、Alt、Shift 键时 ＄18＝HFF，否则＄18＝0。

＊ ＄19、＄1A:执行程序时，暂存将被执行指令的入口地址或字符串值地址，SWAP 中存第一变量名。

＊ ＄1B、＄1C:执行程序中暂存数据，在 SAVE 中作计数器；在系列中断处理程序中为转移程序入口；＄1B 在 READ、INPUT 中存字符串的长度。

＊ ＄1D、＄1E:BASIC 程序变量区首址

＊ ＄1F、＄20:变量区末址

＊ ＄21、＄22:RAM 自由区首址。＄21 单元还作为七页变址寻址器。

＊ ＄24:CONT 指令有效标志。有效时置 HFF；无效时置 0。

＊ ＄25、＄26:POKE 中暂存写入地址

＊ ＄27:暂存变量类型标志

＊ ＄28～＄2A:第一操作数，＄2A 为操作数的种类。在 IF 语句中，存程序运行中对变量的赋值。

＊ ＄2C～＄2E:第二操作数，＄2E 为操作数的种类。在 IF 语句中存变量在变量区的数值。

＊ ＄30:四页自由区首址

＊ ＊ ＄31:屏幕显示标志单元，记录＄2001 软开关的参数。

＊ ＊ ＄32:显示页面对图形库的选择标志单元

* * $33:屏幕显示状态存储单元

$34～$3B:乘、除运算工作单元

$3C～$42:运算结果暂存单元。对于十进制数结果$42=0,从$3D～$41依次存万、千、百、十、个位,$3C为符号位。对于十六进制数$41=0,$3D～$40存四位数。

* $44、$45:程序转向时存目标行号,翻译程序时存行号。

$46:暂存送入$3D及以后单元的运算结果的ASCII码位置。还作为POKE的变址指针。

$47:暂存送入$3D及以后单元的运算结果的ASCII码。按回车键后,暂存本次输入的首行Y座标。

* * $48:光标的当前水平座标

* * $49:光标的当前垂直座标

* $4A、$4B:SPRITE、COLOR中暂存X、Y参数的PPU地址。系列中断程序中为PPU的地址指针。

* $4C:显示页配色区状态标志

$4D、$4E:程序翻译中作为三页的地址指针

* * $4F:背景显示页面在PPU中的起始地址高位。00页为H20、10页为H24。

$50:Caps Lock键工作单元

$52、$53:在间接变址中用于设置转移入口。INPUT中$52用于记录输入字符存于五页的首址。

$54、$55:程序翻译中用于变址寻址

$56、$57:随机数发生器

$58、$59:执行程序时,暂存当前处理指令的地址。

$5A:三页自由区指针。程序翻译时存本语句的长度。

* $5B、$5C:暂存执行程序行的地址

$5D:检查动作卡通状态的计数器

$5F、$60:在执行循环语句、转子语句时暂存栈顶两单元的内容。

* * $61:光标形状存储单元

$62:七页地址指针

* * $63:系统中断处理转向控制器

$64:七页送显的首址指针

$65:PALET标志。PALETB时$65=0、PALETS时$65=1。

* $66:功能键输入标志。按功能键时$66=HFF。

* $67:功能键字符地址指针。高位为功能键编号、低位为字符个数。

* $68:读完功能键和空格键标志。读完时$68=HFF。

* * $69:屏幕滚动控制开关。可上、下移动整幅画面。

$6A:读功能键时暂存键值

$6B、$6C:暂存LOCATE指令中的X、Y参数

* $6D、$6E:读键中,键值与键码换算工作单元。

* $6F:Ctrl键工作单元。按该键时$6F=HFF。

* $70:Shift 键工作单元。按该键时 $70=HFF。
* $71:Alt 键工作单元。按该键时 $71=HFF。
* $72:Caps Lock 键工作单元。按一次该键时 $72=HFF;再按一次 $72=0。
* $73:读 PPU 工作单元

$74:计数器

* $75:键码暂存器
* $76:五页地址指针

$77～$79:连读键功能建立工作单元。其中 $79 为稳键时间控制器。

$7B:连读键标志。连读时 $7B=HFF。

$7C、$7D:DIM 时暂存数组名称。循环语句中暂存循环变量值在变量区的首址。

$7E、$7F:DIM 时暂存数组的两个下标

$80、$81:暂存循环变量的终值

$82、$83:暂存循环步长。DIM 时,$82 存数组维数标志,一维置零,两维置一。

$84、$85:存循环语句的下一行首址。$84 还作为变量类型暂存器,简单变量置 2;串变量置 3;数型下标变量置 H82;串型下标变量置 H83。

$86、$87:暂存循环语句的行号。LIST 时存起始行号。

$88、$89:暂存循环体首址。LIST 时存终止行号。

$8A、$8B:变量名称暂存器

$8C、$8D:暂存当前变量值的存放地址

$8E、$8F:可使用变量区首址

$90:变量类型标志。KEY 时作为字符计数器;STICK、STRIG 时暂存其函数值。

$91、$92:SAVE 中作计数器。$91 还作为 PLAY 中的变址指针。

$98～$9B:程序翻译工作单元。其中 $9A、$9B 还作为五页地址指针和 $09、$0A 地址指针的暂存器。

$9C:COLOR 语句中的配色代码

* * $9D:画面横向座标控制器,可左、右移动整幅画面。

$9E:背景一页禁清标志。

$9F:执行 LOAD 标志。LOAD 时置 0,LOAD? 时置 HFF。

$A4:存动作卡通的编号。

$A5:DEF MOVE 工作单元,暂存卡通块编号的八倍值。

$A6～$A8:动作卡通工作单元

$A9:中断处理中检查动作卡通状态时存卡通编号。

$AB:动作卡通启动标志

$AC:MOVE n 时,置入 HFF。

$AD、$AE:SAVE 中存五页的起始地址。$AD 还作为向 PPU 写入控制器。定义卡通时暂存卡通的符号合成型参数。

$AF、$B0:SAVE 中存五页的结束地址。

$B1:LOAD 工作单元

$B2、$B3:程序转向时存目标行的程序体首址。读 DATA 语句时存数据地址。

$B4:读到 DATA 语句标志,读到时置 $B4=HFF。

$B5:SAVE、LOAD 中的文件名标志,有文件名时置“1”。

$B6、$B8:出错语句的行号。

$B9～$D4:PLAY 工作单元。其中$B9～$BB 存三个系统的节拍参数;$BC～$BE 存 V 参数;$BF～$C1 存 O 参数;$C3～$C4 存音长参数;$C6～$C7 存三个系统的弦乐资料在七页的存储地址;$C9 为演奏标志;$CB～$CD 存三个系统演奏有效标志;$CF～$D1 存 M 参数;$D2～$D4 存 Y 参数。

$D5～$DD:动作卡通工作单元。

* $E0:光标形状(常态)。

* $E1:光标形状(小写字母状态)。

$E4:背景零页标志。

$E5:背景一页标志。

* * $ED～$EF:非屏蔽中断入口。

$F0:显示页面标志。

$F1～$FF:背景音乐、效果音响工作单元。

2、系统一页

$100～$1FF:为堆栈工作区。

3、系统二页

卡通定义区。其中$200～$27F 为显示卡通定义区;$280～$2FF 为动作卡通定义区。

4、系统三页

$300～$3FF:为命令、程序翻译后的存储区。

5、系统四页

$400～$47F:工作中间数据暂存区。

$480～$4FF:功能键定义区。

6、系统五页

$500～$5FF:键盘输入缓存区。

7、系统六页

$600、$619:背景零页使用标志区的开始与结束单元。

$601～$618:背景零页 24 行的使用标志单元。当该行占满时对应单元置入 HFF;否则,置入零。

$61A、$633:背景一页使用标志区的开始与结束单元。

$61B～$632:背景一页 24 行的使用标志单元。当该行占满时对应单元置入 HFF;否则,置入零。

$63D～$644:键盘扫描中的键值暂存区。

$645～$64C:存八个动作卡通的状态。

$680～$69F:存八个显示卡通的 32 块卡通块的序号。

$6C0～$6FF:存八个动作卡通的定义数据,每个卡通占八个单元。

8、系统七页

$700～$7FF:屏幕显示缓冲区和中间数据暂存区。其中$700～$763 为 PLAY 语句工作单元,用于暂存三个系统的弦乐资料。

9、用户程序储存区$703E～$7FFF。

以上凡带有双星号的单元禁止使用；带单星号的单元在作好现场保护后可以使用。

4·1·3 F BASIC 系统的保留字

F BASIC 系统的保留字表位于系统 ROM 区的＄C128～＄C2C4 单元，保留字入口地址表位于＄C2C6～＄C357 单元，各保留字的代码和入口地址见表 4－1。

表 4－1 F BASIC 保留字代码、入口地址一览表

保留字代码	保留字	入口地址	保留字代码	保留字	入口地址
87	SYSTEM	80D7	9A	ON	954D
A2	END	820B	A8	KEY	95AA
91	DATA	8343	94	DIM	964E
95	REM	8343	A6	SAVE	978C
85	THEN	836D	A5	LOAD	97FC
88	TO	836D	84	RESTORE	98E0
89	STEP	836D	93	READ	994D
9B	OFF	836D	A3	PLAY	9A1D
D2	SPC	836D	DB	CSRLIN	A2F3
D3	TAB	836D	CD	FRE	A31A
9D	NEW	8382	DA	POS	A32E
82	RUN	857C	D7	XPOS	A346
86	LIST	85AF	D8	YPOS	A34F
96	STOP	86DA	CF	PEEK	A358
97	CONT	8717	CA	ABS	A369
8B	PRINT	8740	D1	SGN	A379
81	GOSUB	90C6	D0	RND	A39F
83	RETURN	9126	CE	LEN	A3FF
80	GOTO	9146	CB	ASC	A40A
AD	CALL	918C	E1	SCR＄	A41D
8E	PAUSE	9198	DC	CHR＄	A461
AE	LOCATE	91CA	CC	STR＄	A485
9E	POKE	91DF	DD	HEX＄	A4B5
92	IF	9204	D9	VAL	A4BE
99	CLEAR	9226	DE	INKEY＄	A520
8C	FOR	927E	E0	LEFT＄	A574
8D	NEXT	930D	DF	RIGHT＄	A5BD
8F	INPUT	93C2	D4	MID＄	A5FB
AC	SWAP	9496	D5	STICK	A6AA
90	LINPUT	94F6	D6	STRIG	A6B9
AB	CGEN	A6CA	F0	OR	
AF	PALET	A6EA	F1	AND	
9F	CGSET	A746	F2	NOT	
8A	SPRITE	A7F3	F3	≠	
AA	DEF	A887	F4	>=	
A0	VIEW	A932	F5	<=	
98	CLS	AB75	F6	=	
A4	BEEP	B30F	F7	>	

保留字代码	保　留　字	入口地址	保留字代码	保　留　字	入口地址
A9	COLOR	BA4E	F8	＜	
A7	POSITION	C0B6	F9	＋	
A1	MOVE	C0CE	FA	－	
B0	ERA	C0ED	FB	MOD	
9C	CUT	C0F0	FC	÷	
EF	XOR		FD	×	

4·1·4　F BASIC 的显示画面

F BASIC 系统的字符分辨率为 28 列×24 行，位于 6527CPU 系统的显示页面中间，相对于 32 列×30 行画面：上、下各空三行，左、右各空两列，如图 4－1 所示：

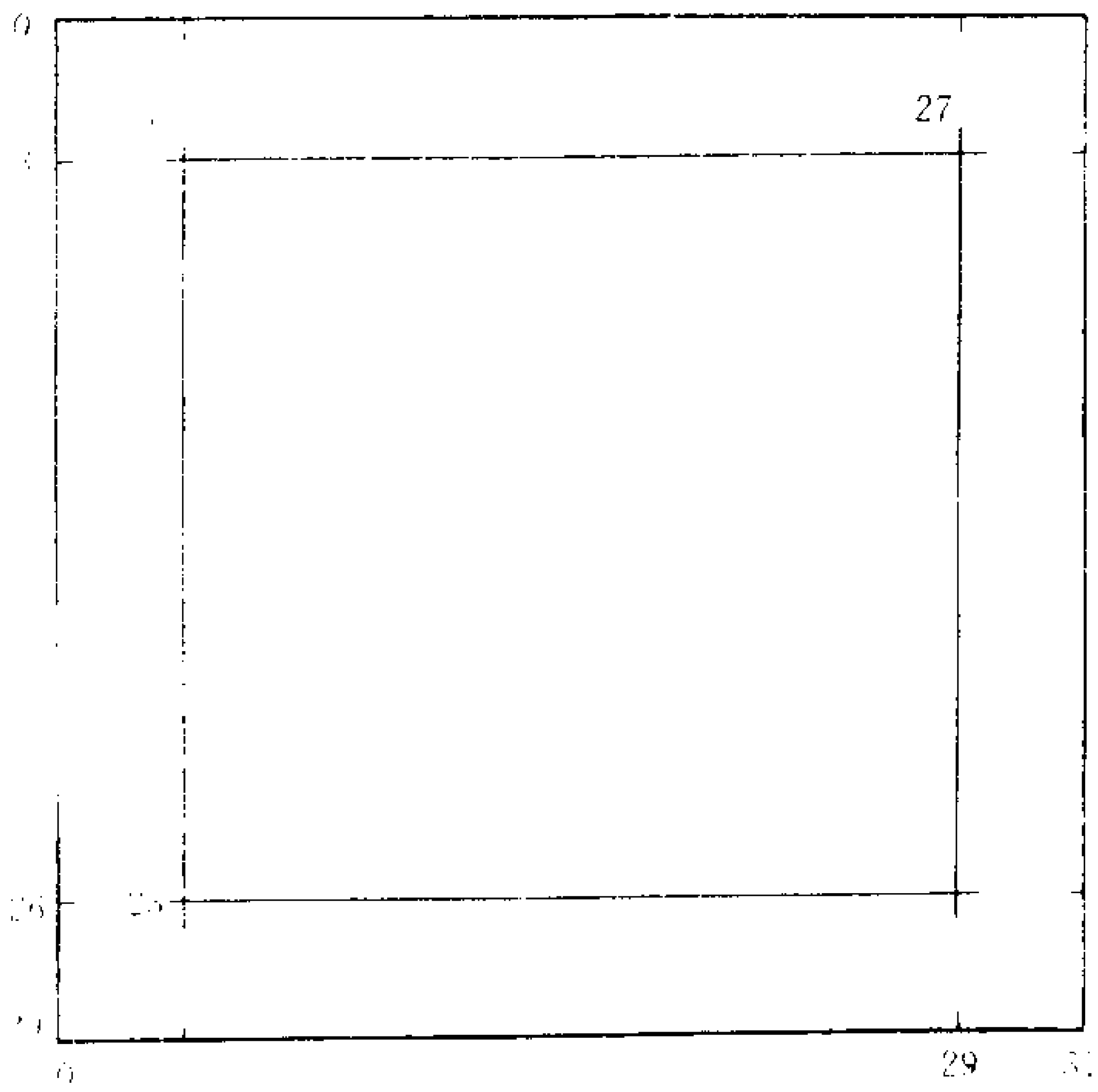


图 4－1　6527 显示画面与 F BASIC 显示画面的对应关系

在 F BASIC 显示画面中，有两种方法显示字符：一种方法是向 PPU 的对应单元写入数据；另一种方法是使用 LOCATE 指令定位、PRINT 指令送显。

4·1·5　F BASIC 程序的输入

F BASIC 程序的输入处理程序位于系统 ROM 区的＄AC03～＄B3AA 约 2K 字节，其中包括屏幕编辑处理、读键处理、清屏处理以及按键声、蜂鸣声的发声控制。所谓程序的输入过程，是指从输入第一个字符起到按回车键为止的全部操作过程。具体工作过程如下：

1、由＄8236 调读键主控程序＄B343，置＄63＝0、＄76＝0，即关闭系统开关和五页地址指针复位。

2、调读键处理程序＄B263，置＄5E＝0，调读键副控程序＄AF88。

3、检查是否按功能键，若按功能键则置功能键标志＄66＝HFF，由功能键扫描处理程序＄AF44 控制读功能键。

4、若没按功能键则调读键程序＄AF9B 读键盘。读出键值后，由键值————键码转换程序＄B02C～＄B08F 处理，把对应的键码存入 A 寄存器返回（非字符键则赋给相应的键码）。若无按键则令 A＝0。

5、判断本次读键是否与键码暂存器＄75 单元中的内容相同（＄75 单元存上一次所读的键码），若不同则转读键初期处理程序＄B2B2；若相同则进入读键中期处理程序＄B274。

6—a 读键初期处理过程

(1) 调＄B254 显示光标；

(2) 对读键计数器＄74 单元置入 H80；

(3) 调读键副控程序＄AF88 读键盘，若有按键则把键码送入＄75，转读键后期处理程序；

(4) 若无按键则令＄74－1 后再读，连读 H80 次无按键则藏光标，重复(2)的操作；

(5) 若仍无按键则转(1)，重复(1)～(5)的操作。

6－b 读键中期处理过程

(1) 检查连读键标志单元＄7B，若＄7B＝HFF 则表示连读键功能已经建立，转连读键处理程序＄B298～＄B2AE 读键，每读一次连读键计数器＄79 减一，当＄79 从 H20 减到零时，认定连读，转读键后期处理。

(2) 若＄7B＝0，表示尚未建立连读键功能，则由＄B27B～＄B28E 处理，反复读键 HFF 次。若仍为同一键则令＄7B＝HFF，建立连读键功能转后期处理。

6－c 读键后期处理

(1) 调＄B25B 藏光标；

(2) 调＄B314 发出击键声；

(3) 令 A＝H75，返回读键主控程序＄B343 执行后续程序。

7、判断按键种类

(1) 若是回车键(A＝0D)则转＄B36E 进行读键输入处理。

(2) 若不是回车键则调＄AC0D 进行判断处理：

· 若是字符键则转＄AC34；

· 若键码是 1、8、9、12(分别为 Del、Tab、Ins 键)则转＄AC25；

· 若不是以上键码则令＄4C＝0，转＄AC25。

8、是字符键的处理

若是字符键则把 A 寄存器的内容送入七页后再送入一个零，置本次传数结束标志；然后令＄64＝0，用 1 触发＄63 单元，显示按键的内容；最后转＄B349 继续读键。

9、按回车键后的处理

按回车键后，由本次输入起始行检索程序＄B36E～＄B37D 处理，检索出本次输入起始行的座标存入＄47 单元，用 2 触发＄63 单元，把本次输入的各字符的 ASCII 码传入五页，并在最后置入三个零作为结束标志。

至此，本次读键过程结束，返回＄8236。

4·1·6 F BASIC 程序的存储格式

我们通过键盘输入的 F BASIC 程序是由指令、变量、数据和各种符号组成的一系列命令。这种形式的命令，CPU 不能直接执行，必须把它翻译成 CPU 能够识别的命令形式才能执行。这一翻译过程称为 F BASIC 程序的编译。编译后的命令形式我们称为 F BASIC 程序的存储格式。不同的工作系统有不同的存储格式(注——BS. 2A 版本与 V3 版本的存储格式也有差别)，F BASIC BS. 2A 版本的存储格式规定如下：

1、所有指令和算符均以表 4－1 中的代码表示。如：在程序存储区，PRINT 指令用 H8B 代替、POSITION 指令用 HA7 代替。显然，这样规定能大量节约存储 F BASIC 程序所占用的内存单元。

2、所有变量和字符均以其 ASCII 码代替。

3、每一个数据占用三个单元，第一个单元表示数据类型(十进制数或十六进制数)，后两个单元存储数值，以十六进制数形式存放，低位在前、高位在后。十六进制数的标志为H11、十进制数的标志为H12。

4、转向语句的目标行号以H0B作为行号标志，每一个目标行号占用三个单元，第一个单元存H0B、第二个单元存行号低位、第三个单元存行号高位。

5、一般把不带行号的程序称为立即执行型程序；把带行号的程序称为暂缓执行型程序。暂缓执行型程序存入$703E以后的内存单元。每一个语句行(一个行号代表的全部程序)以一个零作为结束标志。在一个语句行的存储单元中，第一个单元存本语句行所占用的内存数量(包括第一个单元和最后的结束标志单元)；第二、三两个单元存本行行号，低位在前、高位在后；以后的单元存语句体。如我们输入的第一行程序为：

"10 IF A=&H0312 THEN 100"

在程序区的存储形式如下：

```
703E  11  0A  00  20  92  20  41  F6  11  12  03  20
          行号10        IF        A = &H0312
704A  85  20  0B  64  00  00
     THEN    目标行号100  本行结束标志
```

6、程序区以连续的三个零为结束标志。程序区首址存于$05、$06单元，末址存于$07、$08单元。

7、紧跟在程序区之后的是变量区。变量区的格式如下：

简单变量：

每个简单变量占五个单元。第一个单元为变量类型标志，置入2。第二、三两个单元存变量名，单字符变量名时第三个单元置入零。对于多字符变量名仅取前两个字符。最后两个单元存变量值，变量值取值范围为－32768～＋32767，负数用补码表示。

串型变量：

每个串型变量占用36个单元。第一个单元存串变量的类型标志，置入3；第二、三个单元存串变量名；第四个单元存串变量的长度；第五～第三十五个单元存串变量值；最后一个单元置零作为串变量的结束标志。

下标变量：

数型下标变量标志为H82。

串型下标变量标志为H83。

一维数组标志为零，两维数组标志为1。

定义一个两维数组占用六个单元(一维数组占用五个单元)。其中：第一个单元存数组标志；第二、三两个单元存数组名；第四个单元存数组维数标志；第五个单元存第一下标；第六个单元存第二下标(一维数组无此单元)。

紧跟数组定义单元之后的是下标变量值存储单元。每个数型下标变量占用两个单元；每个串型下标变量占用33个单元；其中第一个单元存串长、第三十三个单元存入零作为串型下标变量的间隔标志，中间三十一个单元存串变量值。

循环变量：

每一个循环变量占用五个单元。第一个为变量类型标志恒置入2；第二、三两个单元存循环变量名；第四、五个单元存循环变量的当前值。

变量名最多允许取 255 个字符，但在变量区中只存前两个字符，即只有前两个字符有效。

4·1·7 F BASIC 程序的编译

所谓程序的编译，是指把用户输入的 F BASIC 程序，按程序存放格式翻译成相应的十六进制数据存入程序暂存区＄300～＄3FF，然后再判断程序的执行类型，属于立即执行型则直接运行程序；属于暂缓执行型则把它存入用户程序区＄703E 以后的内存单元。

编译处理程序位于 ROM 区的＄8236～＄8257 和＄83B1～＄8515。工作过程如下：

1、由＄825F 调＄8A32，对＄09、＄0A 单元置入初始地址＄500。

2、调＄8CD4 读出第一个数据，调＄8CDD 对数据进行判断：

若是数字则表示当前输入的程序是暂缓执行型(有行号)，因而调＄83B1 进行暂缓型程序的翻译；

若不是数字则表示当前输入的程序属立即执行型，故转＄825F 调＄8E59 进行立即执行型程序的编译。

3、立即执行型程序的编译

编译程序位于＄8E59～＄8FF2，编译结果存入＄300 以后的单元中。

(1) 五页地址指针为＄09、＄0A，三页的地址指针为＄4D、＄4E。初值：＄09、＄0A 指向＄500；＄4D、＄4E 指向＄300。

(2) 从五页(＄500)取出第一个数据进行编译判断(＄8E61～＄8FF2)：

· 若是"?"则转＄8EAD 直接把 PRINT 指令的代码 H8B 送入三页；

· 若是单、双引号则转＄8EB3 令＄99＝0 或转＄8EB6 令＄99＝H22，然后把单、双引号的 ASCII 码送入三页而结束编译；(由于单引号为 REM 的简写，属非执行型语句；而开始就是双引号的程序也是非执行型的，故不再编译。)

· 若是". "则转＄8E86 进行简写指令的编译；

· 若是"&"则转＄8ED5 进行十六进制数据的编译；

· 若不是以上符号则转＄8CD0 进行第二次编译判断：若是十进制数据则转＄8ECC，把十进制数据转换成十六进制数据存入＄28、＄29 中；若不是数据则调保留字检索程序＄8FAE，检索保留字代码，若是保留字则将代码存入三页后再取以后的数据编译；若不是保留字则判断是否大写字母，若是大写字母则调＄8E4B，把数据直接送往三页；若读出的数据是零则表示五页的程序编译结束，转＄8E46 进行编译结束处理，向三页送入两个零后结束编译。

4、暂缓执行型程序的编译

五页的地址指针仍为＄09、＄0A，初值为＄500。

(1) 调＄83B1 进入编译处理：首先调十进制数据的 ASCII 码转换为十六进制数的处理程序＄8D09，把行号转换成十六进制数据存入＄28、＄29 单元；然后判断行号是否等于 HFFFF，若是 HFFFF 则转＄8370 进行出错处理；取行号后的数据，若是空格(H20)则调空格编译程序＄8CAD，令＄09＋1 后再读；若行号后面的数据是零则表示要删除该行程序，故转＄8440 进行程序删除处理；若行号后的数据不是零则把＄28、＄29 中暂存的行号送入编译行号暂存器＄44、＄45，最后调＄8E59 编译判断处理程序进行编译判断。

(2) 编译判断处理

· 设置三页地址指针初值＄300 于＄4D、＄4E；

· 从五页取数送入 A 寄存器，判断处理过程同立即执行型程序，直到编译到零后结束编

译；

· 令＄4D＋2 作为本程序行的长度存入＄5A 单元；

· 把行号从暂存器＄44、＄45 单元取出送入＄0F、＄10 单元。

(3) 编译的结束处理——送入程序区

· 调＄8448 把程序区首址＄05、＄06 存入＄5B、＄5C，以＄5B、＄5C 为地址指针读数判断程序区是否有语句。若有语句则转非空区的程序存入处理程序＄8415，按行号顺序把新程序行插入适当的位置；若无语句则进入＄848A 进行空区程序存入处理。

· 空区程序的写入过程

首先令程序区末址加本行语句长度得出存入本行程序后的程序区末址存入＄28、＄29 单元；

然后把语句长度数据送入＄703E 单元；

把行号送入 ＄703F、＄7040 单元；

把三页的内容从＄300 开始，以＄5A 的内容为计数基准，依次取出送入程序存储区；

最后向程序区送入两个零作为程序区的结束标志。

· 调＄9288 进行 CLEAR 处理后转＄8236 继续读键盘。

4·1·8 F BASIC 程序的立即执行过程

没有行号的程序为立即执行型程序。只要一次输入的字符不超过 256 个，按回车键后则立即执行。立即执行处理程序位于 ROM 区的＄8262～＄82D6。这时，编译后的程序存入第三页，执行过程如下：

1、对＄0D、＄0E 单元置地址指针＄B144(指向"0"，当执行非执行型语句 REM、DATA 时，结束程序运行。)，对＄09、＄0A 单元置入程序区首址＄300。

2、调＄86D4 扫描 STOP 键，若未按 STOP 键则返回；否则进行 STOP 处理。

3、调＄8A5B 检查堆栈空区，若空区小于 H14 则转"内存不足"出错处理。

4、从三页取数进行判断处理：

· 若 A＝0 则转＄8345 进行非执行语句处理；

· 若 A＝H3A(:)则转＄8275 执行下一语句；

· 若 A＝H27(')则转＄8345 进行非执行语句的处理；

· 若 A≥H80 表示是保留字，则转保留字入口检索处理程序＄8321，从＄C2C6 以后的数据区检索出保留字入口地址存入＄19、＄1A，然后转入相应的保留字处理程序；处理完毕后转＄8275 继续读取程序数据；

· 若 A 是大写字母则调＄9F75 进行变量的判断处理；处理完毕后转＄8275；

· 当执行完最后一条语句后，从＄09、＄0A 指向的单元中读出的数据为零，则转＄8345 非执行语句处理程序，送显程序执行完毕提示符"OK"，令＄24＝0，置 CONT 指令无效；然后返回读键等待状态，等待用户输入新的程序。

4·1·9 F BASIC 程序的暂缓执行过程

程序的暂缓执行由 RUN 命令引导，执行过程如下：

1、执行 RUN 命令，首先从＄300 单元读出 RUN 的 ASCII 码 H82，然后经保留字转向处理程序＄8321 处理转入 RUN 的解释程序入口＄857C。

2、调＄9228 执行 CLEAR 指令，从变量区首址开始连续置入两个零，使变量区复位。

3、调＄A96E 对六页单元初始化；调＄A6DF 对二页单元初始化。

4、令＄1FF＝HFF、＄1FE＝HFF，设置堆栈底标志(当连续从堆栈弹出两个 HFF 时，表示已到栈底)。

5、把 F BASIC 程序区首址＄703E 从＄05、＄06 单元取出存入＄09、＄0A 单元。

6、转＄834D，首先从＄09、＄0A 单元指定地址中(开始时为＄703E)取出第一行程序占用内存单元数量，换算出下一程序行首址存入＄0D、＄0E 单元中；然后取出当前程序行行号存入＄0B、＄0C 单元中；最后令＄09＋3，指向当前程序体首址，转入＄8275 开始执行程序体内容。

7、以后的执行过程同立即执行型程序。执行结束后返回读键状态，等待用户输入新的程序。

4·2　作用于卡通指令的讨论

BS.2A 版本中作用于卡通的指令共有 13 条，分别为

SPRITE ON	SPRITE OFF
DEF SPRITE	SPRITE n,X,Y
SPRITE n	DEF MOVE
POSITION	MOVE
MOVE(n)	CUT
ERA	XPOS
YPOS	

本节将分别讨论它们的执行过程和工作实质。

4·2·1　卡通的定义指令

F BASIC 中有两类卡通：

- 显示卡通，简称显示 SP。
- 运动卡通，简称动作 SP。

顾名思义，显示 SP 是专用于显示场景的 SP。它通过 DEF SPRITE 指令定义、SPRITE n，X，Y 指令确定显示座标、SPRITE n 指令解除定义。若连续改变显示 SP 的显示座标和姿式，可使显示 SP 产生运动效果。但其运动姿式的改变、显示座标的调整、运动速度的控制都要人为编程解决。

动作 SP 专用于运动 SP 的场景。它由 DEF MOVE 指令定义、POSITION 指令确定运动起点座标、MOVE 指令启动卡通运动、MOVE(n) 指令测试运动情况、CUT 指令停止卡通运动、ERA 指令控制卡通消失。同时还可用 XPOS、YPOS 指令随时测定卡通的当前座标。动作 SP 的运动控制程序极为简单，只要对其控制指令置入合适的参数，动作 SP 即可自动运动。至于运动中 SP 姿式的改变、显示座标的调整、运动速度的控制等操作都由电脑自动完成，故程序处理简洁、直观。当设定动作 SP 的运动方向为零时，动作 SP 也可以作为显示 SP 使用。

显而易见，两种卡通可以互相替换，但它们各有特点：显示卡通的运动控制程序复杂，但其显示座标可编程控制，因而显示 SP 可进行大跨度运动和曲线运动；动作 SP 运动控制程序简

单，但它只能以点为单位运动，且只有八个运动方向，故运动形式较单调。明确它们各自的优、缺点，将有助于编程中对两类卡通的灵活使用。

两类SP的定义都由保留字“DEF”引导。“DEF”的解释程序入口在＄A887。执行＄A887程序，判断DEF后的内容：若是H8A（SPRITE）则执行DEF SPRITE解释程序＄A88B～＄A931，定义显示卡通；若是HA1（MOVE）则转＄BFE4，执行DEF MOVE解释程序＄BFE4～＄C09E，定义动作卡通；若是其它数值则转＄836D，显示“SN”出错提示（语法错误）。

以上说明：DEF保留字后只能是SPRITE或MOVE。

4·2·2 显示卡通的定义

定义显示卡通的指令为DEF SPRITE，语句形式为

DEF SPRITE n，（A，B，C，D，E）＝字符表达式

其中“n”为显示SP的编号，“A，B，C，D，E”依次为配色代码、符号合成型参数（卡通的大、小）、显示优先度、X轴反转标志、Y轴反转标志。

一、定义显示SP的说明

1、定义显示卡通的操作就是在二页的＄200～＄27F单元写入对应卡通编号的X、Y座标，SP块在字库中的序号和显示状态。在二页中，每个16×16点阵的SP由四个SP块组成，每一个SP块占用四个内存单元，一个16×16点阵的SP占用十六个单元。各编号显示SP占用二页地址如表4－2。

表4－2 八个显示SP占用二页地址分配表

编　　号	地　　　址	编　　号	地　　　址
0	200～20F	4	240～24F
1	210～21F	5	250～25F
2	220～22F	6	260～26F
3	230～23F	7	270～27F

2、组成显示卡通的四个SP块的排列顺序如下：

SPn1	SPn2
SPn3	SPn4

3、每个SP块的地址分配

以0＃显示卡通为例，四个SP块依次为SP01、SP02、SP03、SP04，地址分配为：

SP01：＄200～＄203　　SP02：＄204～＄207

SP03：＄208～＄20B　　SP04：＄20C～＄20F

4、每个SP块的四个单元的分工如下：

以SP01为例

- ＄200：存显示卡通块的纵座标（Y值，取值范围0～239）；
- ＄201：存卡通块在图库中的序号；
- ＄202：存卡通块的显示状态，其各位含义如下：

D7	D6	D5	D4	D3	D2	D1	D0

D7:Y 轴反转,0:正常,1:上、下颠倒

D6:X 轴反转,0:正常,1:左、右反转

D5:显示优先度,0:卡通零页,1:卡通一页

D1D0:配色号码,00B,01B,10B,11B

· $203:显示卡通块的横座标存储单元(X 值取值范围 0～255)。

5、定义显示 SP 时要用到六页的以下单元

· $63D～$644:用于存放八个显示卡通的符号合成型参数(SP 的大、小,取值 0、1)。

· $680～$69F:用于存放八个显示卡通的 32 个 SP 块的图形序号,每个显示卡通占四个单元。

· 各编号显示卡通占用六页单元的范围如表 4－3。

表 4－3　显示卡通占用六页单元的分配表

显示 SP 编号	符号合成型参数单元	卡通块图形库序号存储单元
0	$63D	$680～$683
1	$63E	$684～$687
2	$63F	$688～$68B
3	$640	$68C～$68F
4	$641	$690～$693
5	$642	$694～$697
6	$643	$698～$69B
7	$644	$69C～$69F

二、定义显示卡通的操作过程

1、执行 $A88B,调 $A73B 卡通编号检查子程序,判断卡通编号是否合法(显示卡通编号小于八为合法)。

2、执行 $A891～$A8A7,把卡通的编号、配色、大小、显示优先度、X 轴反转、Y 轴反转六个参数从程序区读出入栈保存。

3、执行 $A8AE～$A8B4,Y 轴反转参数出栈,令其循环右移两次,移入 D7 位,屏蔽其它七位后存入显示卡通状态参数暂存器 $19 单元。

4、执行 $A8B5～$A8BE,X 轴反转参数出栈,令其循环右移三次,移入 D6 位,屏蔽其它七位,与 $19 单元的内容相"或"后,再存入显示卡通状态参数暂存器 $19 单元。

5、执行 $A8BF～$A8C9,显示优先度参数出栈,令其循环右移四次,移入 D5 位,屏蔽其它七位,与 $19 单元的内容相"或"后,再存入显示卡通状态参数暂存器 $19 单元。

6、符号合成形参数出栈,存入 $AD 单元。

7、执行 $A8CD～$A8D3,配色参数出栈,屏蔽高六位,与 $19 单元的内容相"或"后,再存入显示卡通状态参数暂存器 $19 单元。

8、显示卡通编号出栈存入 $75 单元。

9、执行 $A8D7～$A8F5,令卡通编号左移四次后送入 X 寄存器;然后以 X 变址,令

$200＋X＝00　　　　$200＋X＋4＝00

$200＋X＝00　　　　$200＋X＋4＝00

$200＋X＝$19　　　　$200＋X＋4＝$19

$200＋X＝20　　　　$200＋X＋4＝20

$200＋X＋8＝00　　　　$200＋X＋C＝00

$200+X+8=00　　　　$200+X+C=00

$200+X+8=$19　　　　$200+X+C=$19

$200+X+8=20　　　　$200+X+C=20

以上操作，把 n 号卡通的二页对应单元置入初值。即令 X、Y 座标单元为零，状态单元置入 $19 的内容，卡通块序号单元置入 H20。

10、调 $9C7C 读数判断嵌套子程序，读出 DEF SPRITE 语句等号后的四个 SP 块的序号存入 $780～$783 单元，并由 $400 单元记录 SP 块的数量，$401、$402 单元指向 SP 块序号首地址 $780。

11、执行 $A8F9～$A929，取出 SP 块序号存入 $680+4n ～$680+4n+3 单元(n 为显示卡通的编号)。

12、把存于 $AD 单元中的卡通符号合成型参数送入 $63D+n 单元。

至此，DEF SPRITE 语句全部执行完毕。

三、DEF SPRITE 语句的执行实质

1、把 DEF SPRITE 语句中的显示卡通的 Y 轴反转、X 轴反转、SP 显示优先度、配色代码读出移位相“或”后存入 $19 单元。

2、把与显示 SP 编号 n 对应的二页单元中的 X、Y 座标单元置入零，卡通块序号单元置入 H20，状态单元置入 $19 单元的数值。

3、把 SP 块的序号存入 $680+4n～$680+4n+3 单元。

4、把显示 SP 的符号合成型参数送入 $63D+n 。

4・2・3　显示 SP 的显示与定义的解除

以 SPRITE 保留字引导的指令有四条，分别为开卡通页显示指令 SPRITE ON、关卡通页显示指令 SPRITE OFF、卡通显示指令 SPRITE n,X,Y、显示卡通解除指令 SPRITE n

SPRITE 保留字的解释程序入口地址为 $A7F3。

执行 $A7F3，判断 SPRITE 保留字以后的内容：

1、若为 H9A(ON)则为 SPRITE ON 指令，转开卡通页显示处理程序 $A7C0。把 $33 单元的 D4 位置 1，把 $2001 显示软开关的 D4 位置 1，即打开卡通页显示开关。

2、若为 H9B(OFF)则为 SPRITE OFF 指令，转关卡通页显示处理程序 $A7CC。屏蔽 $33 单元的 D4 位和 $2001 显示软开关的 D4 位，即关闭卡通页显示开关。

3、若其值小于 H80 则调卡通编号审查子程序 $A73B。若编号小于八则视为合法；否则转出错处理程序 $8370，显示“IL”错误提示(叙述错误)。

4、若 SPRITE n 后的数据为 H2C(“，”)则表示为 SPRITE n,X ,Y 卡通显示指令，则执行 $A870 以后的程序。首先取出 X 座标值存入 $19、Y 座标值存入 $1B、编号 n 存入 $75，并从 $63D+n 单元取出显示卡通的符号合成型参数存入 $90，从 $680+4n～$680+4n+3 中取出执行 DEF SPRITE 语句时存入的卡通块序号，送入二页 n 号卡通对应的序号单元；然后，从 $19、$1B 单元取出 X、Y 座标值存入二页显示卡通的座标单元；最后，判断 $90 单元：若 $90=0(表示定义的显示卡通为 8×8 点阵)则结束操作；若 $90=1(表示定义的显示卡通为 16×16 点阵)则调整 X、Y 座标值，分别置入二页中显示卡通的另外三块卡通块的座标单元。

5、若 SPRITE n 后的数据为 H3A(“：”)或零则表示当前指令为解除 n 号卡通的定义，转解除 n 号卡通的定义处理程序 $A7D3。首先令 n 左移四位赋给 X 寄存器；然后以 X 变址把二页

中 n 号显示卡通对应的座标单元、卡通块序号单元全部置入 HEF(卡通页空格),从而使 n 号显示卡通消失。

小　　结

· 执行 SPRITE ON 的实质是:内部寄存器 $ 2001 的 D4 位置 1;

· 执行 SPRITE OFF 的实质是:对内部寄存器 $ 2001 的 D4 位置 0。

· 执行 SPRITE n,X,Y 的实质:是对 n 号显示卡通对应的二页座标单元、卡通块序号单元置入相应的数据。

· 执行 SPRITE n 的实质是:对 n 号显示卡通对应的二页座标单元置入 HEF。

4·2·4　动作卡通的定义

动作卡通的定义指令为 DEF MOVE,其语句形式为

DEF MOVE(n)=SPRITE(A,B,C,D,E,F)

语句中 n 为动作卡通的编号,A～F 依次为卡通种类、运动方向、运动速度、运动量、显示优先度、配色代码。DEF MOVE 的解释程序为 $ BFE4～ $ C09E 。

一、定义动作卡通的几点说明

1、定义动作卡通的操作就是向六页的 $ 6C0～ $ 6FF 单元写入各动作卡通的运动方向、卡通种类序号、运动速度、运动量,然后由中断程序处理把动作卡通的有关数据写入 $ 280～ $ 2FF 单元。

2、每个动作卡通占用二页十六个单元,地址分配见表 4—4。

表 4—4　动作卡通的二页存储单元分配表

编　号	地　　址	编　号	地　　址
0	280～28F	4	2C0～2CF
1	290～29F	5	2D0～2DF
2	2A0～2AF	6	2E0～2EF
3	2B0～2BF	7	2F0～2FF

3、每个动作卡通占用六页九个单元, 地址分配见表 4—5。

表 4—5　动作卡通的六页存储单元分配表

动作 SP 编号	卡通状态存储单元	卡通的其它参数存储单元
0	$ 645	$ 6C0～ $ 6C7
1	$ 646	$ 6C8～ $ 6CF
2	$ 647	$ 6D0～ $ 6D7
3	$ 648	$ 6D8～ $ 6DF
4	$ 649	$ 6E0～ $ 6E7
5	$ 64A	$ 6E8～ $ 6EF
6	$ 64B	$ 6F0～ $ 6F7
7	$ 64C	$ 6F8～ $ 6FF

4、动作卡通的每个状态单元各位的含义如下:

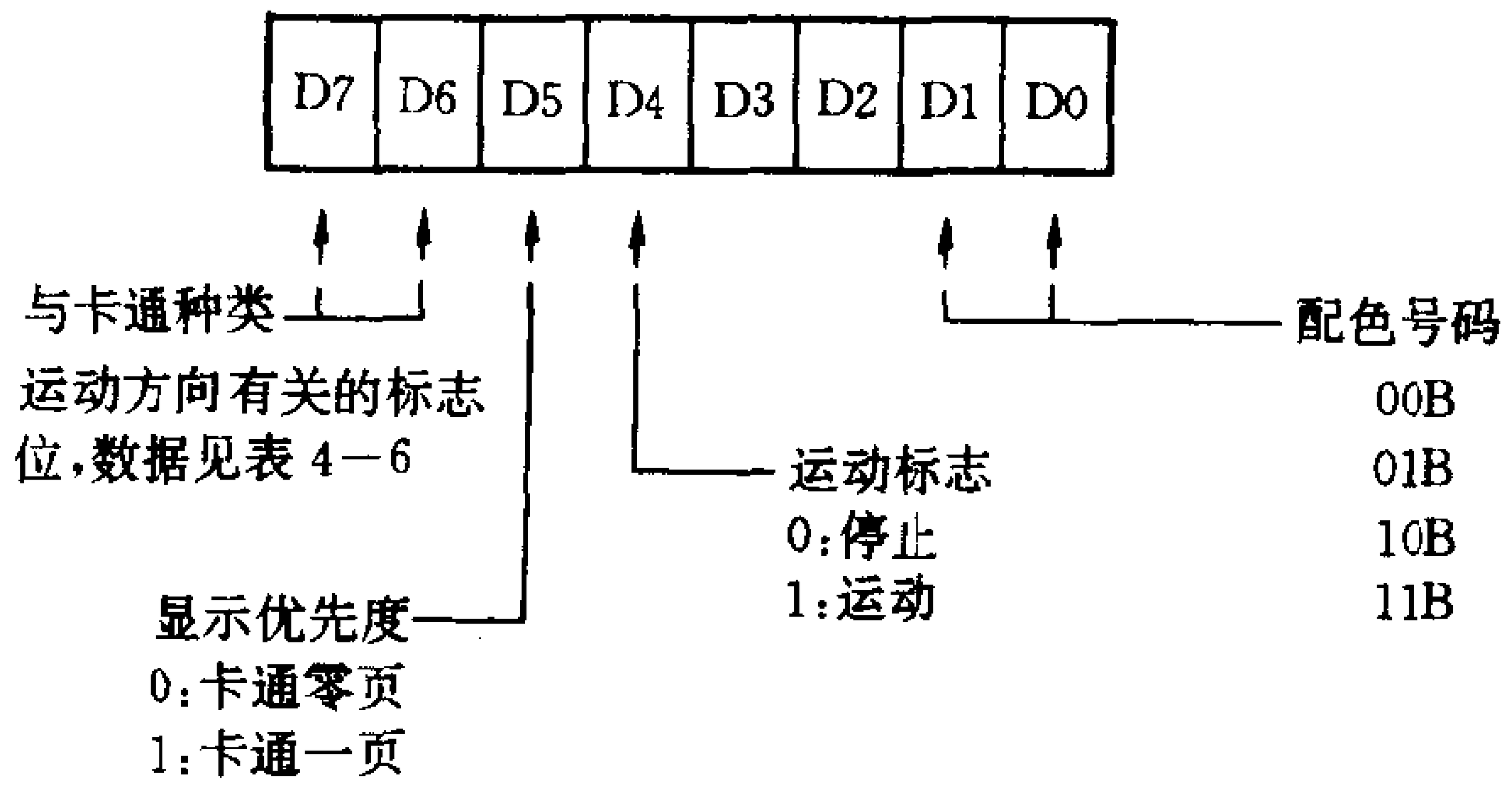


表 4－6　动作卡通状态单元与卡通种类、运动方向的关系

运动方向		马里欧 丽莎		火球 乌龟 企鹅 飞鸟		苍蝇 爆炸 妖怪 螃蟹 太空保垒		汽车 飞鱼 镭射 太空船 卫星杀手	
方向代码	运动方向	D7	D6	D7	D6	D7	D6	D7	D6
0	不动	0	0	0	0	0	0	0	0
1	向上	0	0	0	0	0	0	0	0
2	右上	0	1	0	0	0	1	0	1
3	向右	0	1	0	0	0	1	0	1
4	右下	0	1	0	0	1	1	0	1
5	向下	0	0	0	0	1	0	0	0
6	左下	0	0	0	0	1	0	0	0
7	向左	0	0	0	0	0	0	0	0
8	左上	0	0	0	0	0	0	0	0

5、每个动作卡通使用＄6C0～＄6FF 单元的分工如下:(以 0＃动作卡通为例)

- ＄6C0:动作卡通的运动方向
- ＄6C1:动作卡通的种类序号
- ＄6C2:动作卡通的运动速度计数器
- ＄6C3:动作卡通的运动量计数器
- ＄6C4:动作卡通的运动速度定义值
- ＄6C5:动作卡通的运动量定义值
- ＄6C6:动作卡通的运动起点的 X 座标
- ＄6C7:动作卡通的运动起点的 Y 座标

二、定义动作卡通的操作过程

1、执行＄BFE4,读出动作卡通的编号,调卡通编号审查子程序＄A73B,若 n<8 则编号合法。把编号存入＄A4,令 n 扩大八倍后存入＄A5,有＄A4＝n、＄A5＝8n,并把状态单元＄645＋n 置零。

2、执行＄C011,读出动作卡通的种类序号,存入六页中动作卡通种类序号存储单元＄6C1＋8n。

3、执行＄C01C,读出动作卡通的运动方向,存入六页中动作卡通运动方向存储单元＄6C0＋8n。

4、执行＄C026～＄C04A,根据卡通种类和运动方向从＄BCCF 数据区取出状态数据(对

应于表 4－6)置入＄645＋n。

5、执行＄C050,读出动作卡通的运动速度值存入六页中动作卡通的速度存储单元＄6C2＋8n、＄6C4＋8n。

6、执行＄C059,读出动作卡通的运动量值加一后存入六页中动作卡通的运动量存储单元＄6C3＋8n、＄6C5＋8n。

7、执行＄C06A,读出动作卡通的显示优先度参数,若为零则令状态存储单元＄645＋n 的 D5 位置零;若为 1 则令＄645＋n 的 D5 位为 1。

8、执行＄C084,读出动作卡通的配色代码屏蔽高六位后与＄645＋n 单元的内容、进行“或”运算后再存入＄645＋n。

9、向 X、Y 座标单元＄6C6＋8n、＄6C7＋8n 置入 X、Y 座标初值 H78。

至此,动作卡通的定义操作结束。

三、DEF MOVE 指令的执行实质

执行 DEF MOVE 指令的实质是:向六页中动作卡通的九个单元置入定义参数。其中

＄645＋n 存动作卡通的状态参数。

＄6C0＋8n 存动作卡通的运动方向。

＄6C1＋8n 存动作卡通的种类序号。

＄6C2＋8n、＄6C4＋8n 存动作卡通的速度参数。

＄6C3＋8n、＄6C5＋8n 存动作卡通的运动量参数。

＄6C6＋8n、＄6C7＋8n 存动作卡通的座标初始值。

定义动作卡通只对六页操作。至于把六页的数据写入二页和动作卡通的运动控制,都由 F BASIC 工作系统的中断处理程序完成。这一部分将在中断一节专题讨论。

4·2·5 动作卡通的运动控制与定义解除

动作卡通的运动控制指令有:运动起点定位指令 POSITION、卡通运动启动指令 MOVE、运动停止指令 CUT、令动作卡通消失指令 ERA。

1、动作卡通的运动起点的定位

POSITION 指令的解释程序入口在＄C0B6～＄C0CD。它的执行过程是:首先读出定位动作卡通的编号 n,令 n×8 后送入＄A5;然后读出 X 座标值存入六页与 n 号卡通对应的 X 座标单元＄6C6＋8n;最后读出 Y 座标值存入六页与 n 号卡通对应的 Y 座标单元＄6C7＋8n 中。

2、动作卡通的运动启动

启动动作卡通运动的指令形式为“MOVE n ”。MOVE 保留字的解释程序入口在＄C0CE～＄C0EC。

执行＄C0EC,首先把 n 号动作卡通的状态单元＄645＋8n 的 D4 位置 1;然后令＄AC＝HFF,置动作卡通运动启动标志,供中断程序处理。

3、动作卡通的停动

动作卡通的停动指令为“CUT n”, CUT 保留字的解释程序入口为＄C0F0。执行 CUT n 的操作过程就是屏蔽状态单元＄645＋n 的 D4 位，令 D4 位置零。

4、动作卡通的消失

令 n 号动作卡通消失的指令为 ERA n，ERA 保留字的解释程序入口为＄C0ED。执行＄C0ED,首先屏蔽状态单元＄645＋n 的 D4 位;然后令

$286+16n=F8　　　$284+16n=F8

$288+16n=F8　　　$28C+16n=F8

即令二页中 n 号动作卡通对应的 Y 座标单元置入 HF8,令动作卡通移到卡通显示页之外。

小　结

・执行 POSITION 的实质是:把 n 号卡通的 X、Y 座标值写入对应的六页单元。

・执行 MOVE 的实质是:把 n 号卡通的状态单元 $645+n 的 D4 位置 1,并令 $AC=HFF。

・执行 CUT 的实质是:把 n 号卡通在六页的状态单元 $645+n 的 D4 位置零。

・执行 ERA 的实质是:把二页中与 n 号卡通对应的 Y 座标单元置入 HF8,使卡通移到卡通显示页之外。

4・2・6　动作卡通的测试

动作卡通的测试指令有三条:

・动作卡通的运动状态测试指令为 MOVE(n),当 n 号卡通运动结束时 MOVE(n)=0;若正在运动中则 MOVE(n)=-1。

・动作卡通的当前横座标测试指令 XPOS。

・动作卡通的当前纵座标测试指令 YPOS。

1、动作卡通运动状态的测试

MOVE(n)指令的解释程序入口在 $A29D(注——因 MOVE(n)指令从属于 MOVE 保留字,所以保留字入口表中没有该指令的入口)。

执行 $A29D,首先调 $A3E6 读数判断嵌套子程序,从程序区读出被测试卡通的编号存入 $28 单元;然后检查六页中 n 号卡通的状态单元 $645+n 的 D4 位,若 D4=0 则表示运动结束,令 $400= $401= $402=0,若 D4=1 则表示卡通尚在运动中,于是令 $400= $401=HFF,结束测试。

当需要 MOVE(n)值输出时,由 $8D96 程序处理,若 $400= $401=0 时输出零;若 $400= $401=HFF 时输出 MOVE(n)=-1。

2、动作卡通的横座标测试

动作卡通的横座标测试指令 XPOS 的解释程序入口在 $A346。执行 $A346,首先调 $A33C 读出被测试的动作卡通编号 n,令 n 乘以 8 后送入 X 寄存器,有 X=8n;然后以 X 变址,从 $6C6+X 单元中取出与 n 号卡通对应的 X 座标单元的数据送入 $28 单元;最后转 $A301,把 n 号卡通的 X 座标值送入 $400 单元以备输出。

3、动作卡通的纵座标测试

动作卡通的当前纵座标测试指令 YPOS 的解释程序入口在 $A34F。其测试过程除从 $6C7+X 单元中取数外,其它同 XPOS。

小　结

・执行 MOVE(n)的实质是:检查 n 号卡通状态单元的 D4 位:若 D4=0 则令 $400= $401=0;若 D4=1 则令 $400= $401=HFF。

・执行 XPOS(n)的实质是:读出 n 号卡通在六页中对应的 X 座标值送入 $400 单元,并令 $401=0。

・执行 YPOS(n)的实质是:读出 n 号卡通在六页中对应的 Y 座标值送入 $400 单元,并

令＄401＝0。

4·3 字符函数指令的讨论

F BASIC BS. 2A 版本共有字符函数九个。其中作用于 ASCII 码的函数两个：ASC、CHR＄；作用于数据进制转换的函数三个：VAL、STR＄、HEX＄；作用于字符串的函数四个：LEFT＄、RIGHT＄、MID＄、LEN。

4·3·1 字符 ASCII 码的转换

1、ASC 函数

ASC 函数的功能是求出任一字符的 ASCII 码，其解释程序入口在＄A40A。

执行＄A40A，首先调＄A3F9 括号内变量检索处理程序，从变量区检索出串变量名存入＄28 单元，串变量值位于变量区的首址存入＄29、＄2A 单元；然后从字符串中取出第一个字符存入＄28，转＄A402，令＄29＝＄2A＝0，再转＄A301，令＄400＝＄28、＄401＝＄29、＄402＝＄2A，以备送入变量区或送显。当输出 ASC 函数值时则由 PRINT 语句处理，把存于＄28 单元的字符的 ASCII 码显示在屏幕上。

2、CHR＄函数

CHR＄函数是 ASC 函数的反函数，它的功能是求出与 ASCII 码对应的字符。其解释程序入口在＄A461。

执行＄A461，首先调＄A3E6 子程序读出括号内的 ASCII 码存入＄28 单元，并令＄29＝＄2A＝0；

然后调＄A473 把存于＄28 单元的 ASCII 码送入七页，并由＄29 记录存入七页的地址偏移量，同时令＄28＝1、＄2A＝7；

第三、转＄A305 把＄28、＄29、＄2A 的内容分别存入＄400、＄401、＄402 单元。即由＄400 单元记录存入七页的数据个数，当前＄400、＄401、＄402 单元指出存入七页的地址，以备存入变量区或送显。

小　结

· 执行 ASC 函数的实质是：把串变量中的第一个字符的 ASCII 码取出存入＄400 单元。

· 执行 CHR＄函数的实质是：把 ASCII 码取出存入七页，由＄401、＄402 单元指出 ASCII 码存于七页的地址。

4·3·2 数值、字符串的转换

1、VAL 函数

VAL 函数的作用是把字符串转换成数值。它的特点是：若字符串中的第一字符是字母则函数值为零；若字符串中的第一字符是十进制数字则把字符串中从第一字符向后符合十进制规则的字符转换为数值；若字符串中的第一字符是“&”则把后面符合十六进制数据规则的字符转换为数值；若字符串中的第一字符是“－”则把后面符合转换要求的字符转换为负数。如：

A＄＝“12AB”　　B＄＝“A12345”

C＄＝“&AHI34”　　D＄＝“－123AB”

则有

VAL(A$)=12　　　VAL(B$)=0

VAL(C$)=10　　　VAL(D$)=-123

VAL 函数的解释程序入口在$A4BE。

执行$A4BE:首先调字符串读出判断复合处理子程序$A3F9,检查括号中的内容是串变量还是字符串。若是串变量则把串长存入$28、串值首址存入$29、$2A单元;若是字符串则把字串读出存入七页,把串长存入$28、串值位于七页的首址存入$29、$2A单元。

第二、把$28单元的串长送入$74单元,以$74为计数器,从$29、$2A指定的串值地址中读出串值存入$500以后的单元,读数完毕再向五页置入一个零作为字符串结束标志。

第三、调$A4FE检查字串内容。从字符串的第一个字符开始判断:若是"&"则把后面符合十六进制数据规则的字符转换为十六进制数值存入$28、$29单元;若字符串中的第一字符是"0~9、-、&"以外的字符则令$28=$29=0;若第一字符是"-"则把后面符合转换要求的字符转换为负数;若第一字符是阿拉伯数字则把符合十进制规则的字符转换为十六进制数值存入$28、$29单元中。

第四、把$28、$29单元的数据存入$400、$401单元以备存入变量区或输出显示。

2、STR$ 函数

STR$函数的功能是改变括号中的十进制数据的性质,使其变为字串。其解释程序入口在$A485。

执行$A485:首先调$A3EC读出括号中的数值存入$28、$29单元。

第二、调$8D96把$28、$29单元中的数据转换为对应的ASCII码存入$3D~$41单元。

第三、调$8CEB对输出结果数据整形,删除输出结果前面的零。

第四、把输出结果从$3D~$41中取出存入七页。

第五、转$A4A4把字串长度存入$28单元,把七页的字串首址写入$29、$2A单元。

第六、转$A469置字符串标志$27=3,转$A305把字符串长度存入$400单元,字符串首址存入$401、$402,以备输出送显或写入变量区。

3、HEX$ 函数

HEX$函数的功能是把括号中的十进制常数转换为十六进制数据。其解释程序入口在$A4B5。

执行$A4B5:首先调$A3EC读出括号中的十进制数值存入$28、$29单元(数据形式为十六进制)。

第二、调$8DD8数据→ASCII码转换程序,把$28、$29中的十六进制数据转换为ASCII码(每位一个、共四个)存入$3D~$40单元中,并令$41=0。

第三、转$A48B进行STR$函数的第三~第六步处理。

小　结

· 执行VAL的实质是:把字符串中符合十进制或十六进制的字符转换成十六进制数据存入$400、$401单元。

· 执行STR$的实质是:把转换为字符的ASCII码存入七页,由$400记录串长、$401和$402记录字符串存入七页的首址。

· 执行HEX$的实质是:把括号中的十进制数据转换为十六进制数据后,以ASCII码的形式存入七页,由$400单元记录串长,$401、$402单元指出存于七页的首址。

4·3·3 字符串的组字和测试

1、LEFT＄ 函数

LEFT＄函数的功能是：从字符串 A＄的左边取出指定个数的字符组成子字符串。语句形式为

B＄＝LEFT＄（A＄，n）

其解释程序入口在 ＄A574。

执行＄A574：调＄A555 括号多内容处理子程序，首先调＄A3F9，从变量区查出 A＄串变量，把串长存入＄28，串值首址存入＄29、＄2A。

第二、把＄28～＄2A 的内容入栈保护。

第三、调＄A59E 子程序取出逗号后的数据存入＄28～＄2A。

第四、字符串首址出栈存入＄19、＄1A 单元，串长存入＄4A 单元。

第五、比较欲取字符个数和串长度，若串长小则令＄28＝＄4A。

第六、把字串首址送入＄29、＄2A，转＄A305 把＄28～＄2A 的内容送入四页。欲取串长存＄400、串变量值首址存入＄401、＄402。

2、RIGHT＄ 函数

RIGHT＄函数的功能是：从字符串 A＄的右边取出 N 个字符组成子字符串。语句形式为

B＄＝RIGHT＄（A＄，n）

其解释程序入口在＄A5BD。

执行＄A5BD：首先进行 LEFT＄函数的第一～第五步处理；然后调整＄19、＄1A 数值，使其指向 A＄中后 n 个字符的开始位置；最后从串变量区取出新字符串（子串）存入七页，由＄28 单元记录新串长，＄29、＄2A 记录新串在七页的地址，再转＄A305 把＄28～＄2A 的内容送入四页，令＄400 存新串长、＄401 和＄402 存新串首址。

3、MID＄ 函数

MID＄函数比 LEFT＄、RIGHT＄函数使用灵活，语句形式为

B＄＝MID＄（A＄，n，m）

功能是从 A＄的第 n 位取 m 个字符。MID＄函数的解释程序入口在＄A5FB。

执行＄A5FB：首先调＄A3F9 从变量区检索 A＄串变量，查到后把串长存入＄28 单元，字串首址存入＄29、＄2A 单元，返回后把＄28～＄2A 中的数值入栈保护。

第二、调＄A59E 把逗号后的 n 值读出存入＄28、＄29 单元，检查判断＄28、＄29 单元的数值，若＄29≠0 或＄28＝0 或＄28≥H20 都进行出错处理；若＄28＜H20 则令＄28－1 后，其值入栈。

第三、调＄A59E 把逗号后的 m 值读出存入＄28、＄29 单元，检查判断＄28、＄29 单元的数值，判断过程同 n，若＄28＜H20 则把＄28 中的 m 值送入＄1C 单元。

第四、n 值出栈存入＄1B。字串 A＄首址出栈存入＄19、＄1A。串长 l 出栈与 n 比较。若串长≥n 则令串长－n，再比较 n 位后的字符数量与 m 的大小。若有串长－n＞m 则说明 A＄字串中的字符满足在 n 位取 m 个字符的要求；若串长＜n 则转＄A66E 调＄A2BA，令＄28＝＄29＝＄2A＝0，使新串为空串；若串长－n≤m 则转＄A669，令＄1C＝串长－n，即令所取新串为 A＄中 n 位后的全部字符。

第五、从 A＄的第 n 位开始取出 m 个字符送入七页。由＄29、＄2A 指出新串的首址，＄28

记录新串的长度。

第六、转＄A305 把新串长存入＄400，新串首址存入＄401、＄402。

4、LEN 函数

LEN 函数的语句形式为 A＝LEN(A＄)，功能是测量 A＄的字符个数(串长)。LEN 函数的解释程序入口在＄A3FF。

执行＄A3FF：首先调＄A3F9 把 A＄的串长存入＄28，串址存入＄29、＄2A。

第二、调＄A2BE 令＄29＝0、＄2A＝0。

第三、转＄A301 把存于＄28 中的串长送入＄400，令＄401＝＄402＝0，以备输出或存入变量区。

小　结

・执行 LEFT＄指令的实质是：把新串长存入＄400、串首址存入＄401、＄402 单元。

・执行 RIGHT＄、MID＄指令的实质是：把新串取出存入七页。由＄400 记录串长，＄401 和＄402 记录新串在七页的首址。

・执行 LEN 指令的实质是：从变量区查出串长存入＄400 单元，并令＄401 和＄402 单元都等于零。

4・4　作用于背景显示页指令的讨论

作用于背景显示页的指令有八条：分别为

清屏指令 CLS、调背景一页画面指令 VIEW、输出定位指令 LOCATE、输出指令 PRINT、显示页使用图形库区号设置指令 CGEN、光标坐标测试指令 POS、CSRLIN、画面测试指令 SCR＄

4・4・1　CLS 指令的执行过程

CLS 指令的作用是清除背景页屏幕显示的内容。解释程序入口在＄AB75。

执行＄AB75：首先令＄63＝H0C，关显示页，令＄2000＝＄32 以关闭中断；然后设置与 PPU 通讯的首地址，把选定页面的 PPU 显示单元全部置入 H20(空格)以清屏；第三、用＄9D、＄69 单元的内容分别触发＄2005，使 F BASIC 画面定位；第四、取＄32 单元的内容与 H80 进行“或”运算后送入＄2000 寄存器，以开中断；第五、令＄63＝B 开背景页显示；第六、令＄48＝＄49＝＄69＝＄9D＝0，光标定位于屏幕左上角；第七、令屏幕显示行使用标志单元＄601～＄619 复位(全部置入零)。

若调＄AB75 则清背景零页；

若调＄AB71 则清背景一页。

4・4・2　VIEW 指令的执行过程

VIEW 指令的功能是调出 BG GRAPHIC 绘制在背景一页上的画面。解释程序入口在＄A932。

执行＄A932：首先调＄AB75 清背景零页；然后令＄63＝H0C，关显示；第三、把 PPU 的＄2400～＄27FF 的内容转存到＄2000～＄23FF 的对应单元中；第四、用＄9D、＄69 分别触发＄2005 软开关，使 F BASIC 的显示画面定位；第五、令＄63＝H0B，开显示。

4·4·3 LOCATE 指令的执行过程

LOCATE 指令形式为：

LOCATE X，Y

功能是从(X,Y)座标开始输出打印。解释程序入口在＄91CA。

执行＄91CA：首先调＄9C6D 读出 X 座标值存入＄6B；然后调＄9C6A 读出 Y 座标值存入＄6C；最后转＄ACAA 把＄6B 单元的内容送入＄48、＄6C 单元的内容送入＄49，以对光标定位。

4·4·4 PRINT 指令的执行过程

PRINT 指令的解释程序入口在＄8740。

执行＄8740：第一、调＄9C82 读数子程序进行读数判断：若是数值变量则把变量值存入＄400、＄401 单元；若是字符变量则把串长存入＄400、串首址存入＄401 和＄402 单元；若是字符串则把字符串存入七页，并由＄400 记录字符串长度(字符个数)，＄401 和＄402 记录串变量的首地址。

第二、当输出数值变量时，则执行＄874B：首先把数值存入＄28、＄29 单元；然后调＄8D96 把＄28、＄29 中的十六进制数据转换为十进制数，以 ASCII 码的形式存入＄3D～＄41 单元，各数位对应单元如下：

数位：	万	千	百	十	个
单元：	＄3D	＄3E	＄3F	＄40	＄41

最后，调＄865F 把＄3D～＄41 单元的内容送入七页，令＄63＝1，送显结果。再转＄8794 进行后续输出项的处理。

第三、当输出字符变量时，首先把串变量地址置入＄19、＄1A 单元；然后以＄400 为计数器，把串变量值送入＄500 以后的单元，并对最后一个单元置零；最后调＄87B7 从五页＄500 单元开始取数送入七页，令＄63＝1 送显。再转＄8794 进行后续输出项的处理。

第四、后续处理程序＄8794 的操作如下：取数判断，若后续项为 H3A(:)则表示输出结束，故转＄87B4 令光标下移一行(即换行)；若后续项是 H3B(;)则转＄87A6 读数据，以紧凑格式输出打印；若后续项是 H2C(,)则令 A＝9 调＄AC0D 子程序进行定格打印处理；若后续项不是以上符号则转＄8740 进行下一项输出处理。

4·4·5 CGEN 指令的执行过程

CGEN 指令的语句形式为

CGEN (n)

功能是选定背景页、卡通页的图形区号。解释程序入口在 ＄A6CA。

执行＄A6CA：首先调＄9C6D 子程序读出参数 n；然后只取参数 n 的 D0、D1 位，左移三次后存入＄74 单元；最后令＄32 与 H7E 进行“与”运算后再与＄74 进行“或”运算，把运算结果存入＄32，并把＄32 单元的最高位置 1 后送入＄2000。

CGEN 指令与＄32、＄2000 单元置值的对应关系如表 4－7。

表 4—7 CGEN 指令的执行实质

CGEN 参数	置入＄32 单元的数值	置入＄2000 单元的数值	卡通页选定的图库区号	背景页选定的图库区号
0	00	H80	Ⅰ	Ⅰ
1	08	H88	Ⅱ	Ⅰ
2	H10	H90	Ⅰ	Ⅱ
3	H18	H98	Ⅱ	Ⅱ

4·4·6 POS 指令的执行过程

POS 指令的功能是测试光标目前的水平座标位置。解释程序入口在＄A32E。

执行＄A32E：首先调＄A3EC 读出括号中的零存于＄28 单元中；然后把＄48 单元的数值送入＄28，并令＄29＝0 后转＄A363；最后转＄A301，把＄28、＄29 单元的内容送入＄400、＄401 以备存入变量区或送显。

4·4·7 CSRLIN 指令的执行过程

CSRLIN 指令的功能是测试光标目前的垂直座标位置。解释程序入口在＄A2F3。

执行＄A2F3：首先把＄49 的内容送入＄28 单元，令＄29＝0；然后把＄28、＄29 的内容送入＄400、＄401 单元以备送入变量区或输出送显。

4·4·8 SCR＄ 指令的执行过程

SCR＄ 指令的语句格式为

SCR＄(X，Y) 或 SCR＄(X，Y，n)

功能是求出(X，Y)座标处的图形数据或配色代码(参数 n＝1 时求配色代码；不带参数 n 或参数 n＝0 时求图形代码)。解释程序入口在＄A41D 。

执行 ＄A41D：

第一、调＄ACA1 把光标的当前座标值存入＄6B、＄6C 单元保存。

第二、调＄A3E6 读出给定的 X 座标值存入＄48，调＄A59E 读出给定的 Y 座标值存入＄49。

第三、读取后面的数据，若为 H29(“)”)则表示执行 SCR＄(X，Y)指令，于是进行图形数据的测试处理；若不是 H29 则检查数据的大小，若 n＝0 则转图形测试；若 n＝1 则进行配色代码的测试。

第四、配色代码的测试：首先调＄BA37 根据＄48、＄49 的座标值，取出对应 PPU 配色单元的配色代码存入＄9C 中；然后把配色代码送入＄28，恢复＄48、＄49 单元中光标坐标的原值，再转＄A464，把读出的配色代码送入四页以备送入变量区或输出送显。

第五、图形数据的测试：执行＄A44F，首先调＄ABA6 计算出＄48、＄49 座标对应的 PPU 地址存入＄4A、＄4B，然后令＄63＝5 读出对应地址的图形数据存入＄28 中，恢复＄48、＄49 单元中光标坐标的原值；最后转＄A464 把＄28 单元的内容送入四页以备输出。

小　结

· CLS 指令的执行实质是:对背景零页的 PPU 显示单元置入 H20。

· VIEW 指令的执行实质是:把 PPU 的 $ 2400～$ 27FF 的内容送入 $ 2000～$ 23FF。

· LOCATE 指令的执行实质是:把 X、Y 值送入 $ 48、$ 49 单元。

· PRINT 指令的执行实质是:把输出的数据转换为可显示的 ASCII 码送入七页、输出的字符的 ASCII 码送往七页,然后令 $ 63＝1 显示。

· CGEN 指令的执行实质是:对 $ 2000 的 D4、D3 位置入相应的数值。

· POS 指令的执行实质是:读出 $ 48 单元的内容送入 $ 400。

· CSRLIN 指令的执行实质是:读出 $ 49 单元的内容送入 $ 400。

· SCR $ 指令的执行实质是:测试图形时,计算出指定单元的 PPU 地址,读出该地址的图形数据送入 $ 400 单元;测试配色代码时则计算出指定单元对应的 PPU 配色地址,读出该地址对应的配色代码送入 $ 400 单元。

4·5　转向指令的讨论

F BASIC 的 BS. 2A 版本中与转向有关的指令共有 11 条。分别为:

· 无条件转向指令	GOTO
· 调用子程序指令	GOSUB
· 子程序返回指令	RETURN
· 数据区恢复指令	RESTORE
· 开关转向指令	ON　N　GOTO
· 开关调子程序指令	ON　N　GOSUB
· 开关子程序返回指令	ON　N　RETURN
· 开关数据区恢复指令	ON　N　RESTORE
· 开关程序运行指令	ON　N　RUN
· 条件转向指令	IF ……… THEN
	IF ……… GOTO
	IF ……… THEN GOTO

· 机器语言程序调用指令CALL

其中条件转向指令的三种形式功能相同,故等同为一条。下面分别讨论。

4·5·1　程序的无条件转向

GOTO 指令的功能是控制程序无条件地转移到 GOTO 语句规定的行号继续执行。GOTO 语句的解释程序入口在 $ 9146。

执行 $ 9146:首先调 $ 9172 检索转向的目标行号,若检索到则直接返回;若没检索到则进行出错处理;然后取出目标行的下一行程序首址存入 $ 0D、$ 0E,目标行号存入 $ 0B、$ 0C,目标行程序体首址存入 $ 09、$ 0A,执行程序体的各语句。

4·5·2　子程序的调用和返回

GOSUB 指令的功能是调用、执行子程序,当执行到 RETURN 语句时返回原调用程序。

GOSUB 的解释程序入口在 $ 90C6，RETURN 的解释程序入口在 $ 9126。

1. 栈顶两个单元的内容出栈，分别存入 $ 5F、$ 60 单元。调 $ 9172 子程序，从程序区 $ 703F 开始检索目标行号，若检索到则直接返回；否则进行出错处理。

2. 调 $ 9598，取出行号后的数据，若是 H3A 则直接返回；若是 H2C 则表示后面还有行号，令 $ 09+1 后返回。

3. 把当前的程序地址、行号、下一行程序首址入栈保护，并令 00、HFF 入栈作为 GOSUB 现场数据栈顶标志。最后令 $ 5F、$ 60 的内容入栈。

4. 取出目标行号的下一行首址存入 $ 0D、$ 0E 单元，取出目标行号存入 $ 0B、$ 0C 单元，取出目标行程序体首址存入 $ 09、$ 0A 单元，开始执行子程序。

5. 当执行到 RETURN 语句时，首先检查后面是否有行号。若没有行号则令原入栈保护的现场数据出栈；若有行号则令栈顶两单元出栈，GOSUB 现场数据栈标志 HFF、00 出栈，令栈指针 S 加六后，再把栈顶两单元内容入栈；然后进入 GOTO 语句的处理程序，转向处理 RETURN 后的目标行号程序。

4・5・3　数据区的恢复

RESTORE 指令的功能是恢复 READ 语句的数据区，当 RESTORE 后无行号时，则把数据区指针恢复到第一个 DATA 语句；若有行号则恢复到指定行号的 DATA 语句。其解释程序入口在 $ 98E0。

1. 若 RESTORE 后无行号则转 $ 98FA，令 $ B2、$ B3 指向程序区第一行程序体首址，令 $ B4=0 后返回。使执行 READ 语句时从第一程序行开始检索 DATA 语句。

2. 若 RESTORE 后有行号则调 $ 98E2，检索到目标行号将其存入 $ 5B、$ 5C 单元中；然后令 $ 5B、$ 5C 的地址指针加三存入 $ B2、$ B3，使 $ B2、$ B3 指向目标行程序体首址。令 $ B4=0 后返回。

4・5・4　开关转向指令的讨论

开关转向指令的特点是根据开关参数的取值顺序转入多个程序分支。其解释程序入口在 $ 954D。

1、调 $ 9C6D 子程序读出开关参数 N 存入 $ 400 单元。

2、调 $ 8CD4 读出开关出参数后的保留字代码入栈。

3、读取转向行号进行判断：

(1)若没有转向行号则直接返回。

(2)若开关参数为 1 则转 $ 9576。令保留字代码出栈，若保留字代码大于 H85 则出错；若小于 H85 则查出保留字的入口地址存入 $ 19、$ 1A，然后转入相应的保留字解释程序，处理完毕后返回。

(3)若开关参数大于 1 则令其减一、$ 09 单元加三，跳过第一个目标行号。

(4)反复执行 3，直到开关参数递减到 1 时再执行 2。

4・5・5　机器语言子程序的调用

CALL 指令的功能是调用机器语言子程序。其入口地址在 $ 918C。

执行 $ 918C：首先调 $ 9C82，取出 CALL 后的地址数据送入 $ 400、$ 401 单元；然后转向

$400、$401 指向的地址执行机器语言子程序。执行完毕后返回断点地址，执行后续程序。

4·5·6 程序的条件转移

程序的条件转移由 IF 指令引导。IF 的解释程序入口地址在 $9204。执行过程如下：

1. 调 $9C82，判断 IF 后的内容，根据设定的条件，对 $400、$401 单元设定相应的数值。

2. 令 $400 与 $401 的内容相“或”，检查结果，若 $400 OR $401=0 则说明条件不成立，跳过 IF 语句行，执行下一行程序；若 $400 OR $401≠0 说明条件成立，则执行 THEN 后的程序。

3. 若 THEN 以后是行号则转 GOTO 处理程序；若是其它语句则返回读数判断程序，执行 THEN 以后的程序。

小　结

· GOTO 指令的执行实质是：把目标行号存入 $0B、$0C，下一行程序首址存入 $0D、$0E，目标行程序体首址存入 $09、$0A。

· GOSUB 指令的执行实质是：保护现场，进行 GOTO 指令的处理。执行到 RETURN 后结束。

· RETURN 指令的执行实质是：若 RETURN 后无行号则恢复现场，执行断点后的程序；若有行号则进入 GOTO 指令的处理，返回到目标行号执行程序。

· RESTORE 指令的执行实质是：令 $B4=0，令 $B2、$B3 指向 DATA 语句体首址。

· ON 指令的执行实质是：根据开关参数 N 值，转向对应的次序行号，进入相应的处理程序入口。

· CALL 指令的执行实质是：令程序转向 CALL 指令的程序地址执行程序。

· IF 指令的执行实质是：根据条件对 $400、$401 赋值，若 $400 OR $401=0，说明条件不满足，跳过 IF 语句行；若 $400 OR $401≠0，说明条件满足，执行 THEN 后的程序。

4·6 作用于配色与发声指令的讨论

作用于配色的指令有四条：分别为：选定显示页面配色面板代码指令 CGSET、设定配色代码指令 PALETB、PALETS、背景画面染色指令 COLOR、作用于发声的指令有两条：发蜂鸣声指令 BEEP、音乐旋律演奏指令 PLAY。

4·6·1 配色面板代码的选择

F BASIC 有两类配色面板：背景页配色板、卡通页配色板。在各自的配色板中背景页有两个面板，卡通页有三个面板，编号分别为 0、1 和 0、1、2。每个配色面板包含四个配色代码，编号为 0～3。每个配色代码又由四个颜色代码组成，其中第一个颜色代码为底背景配色，后三个颜色代码为背景图形块或卡通图形块的配色。可见，F BASIC 系统的配色组合是极为方便和复杂的。CGSET 指令是专为选择背景页、卡通页的配色面板而设置的指令。其语句形式为

CGSET M，N

功能是选定两个显示页面的配色面板，M 为背景配色面板编号、N 为卡通配色面板编号。CGSET 指令的解释程序入口在 $A746。执行过程如下：

执行 $A746：首先判断 CGSET 指令后的第一个字符是否是 H2C，是则转 $A76F 进行卡

通页面板的配色处理("CGSET,N"的语句形式表示背景配色面板按系统设定,只改变卡通页配色面板。);若不是 H2C 则调＄9C6D 读出背景配色面板代码 M,根据 M 值从＄C3A5 数据区查出配色数据区首址存入＄52、＄53 单元。

第二、调＄A78A 配色代码定义数据传送子程序,把背景配色面板中 M 号代码的各颜色代码送入 PPU 配色区＄3F00～＄3F0F。

第三、调＄9C64 取出卡通页配色面板代码 N,从＄C3A9 数据区读出 N 号卡通页配色面板的数据首址存入＄52、＄53 单元。

第四、调＄A78A 把卡通页配色数据送入 PPU 配色区＄3F10～＄3F1F。

4·6·2 配色代码的重新设定

PALETB、PALETS 指令是用户自行设定配色代码的颜色组合专用指令。指令形式为

$$\text{PALET}\left(\begin{matrix}\text{B M}\\ \text{S N}\end{matrix}\right),\text{C1},\text{C2},\text{C3},\text{C4}$$

其中"M"为背景页的配色代码,"N"为卡通页的配色代码,取值均为 0～3。

PALET 保留字的解释程序入口在＄A6EA。其定义过程如下:

1..对＄65 单元置入 PALET 指令类型标志,若为 PALETB 则令＄65=0;若为 PALETS 则令＄65=10。

2.调＄9C64 读出配色代码 M(或 N),检查合法后入栈保护。

3.向＄02 单元置入 4,以＄02 单元为计数器,以＄7C 单元变址(＄7C 的初值为 0)反复调用＄9C6A 读数子程序,读出 PALETB 或 PALETS 以后的四个参数存入＄3D～＄40 单元中。

4.M 或 N 值出栈送入＄700,C1 送入＄701、C2 送入＄702、C3 送入＄703、C4 送入＄704。

5.检查＄65 单元:若＄65=0 则令＄63=6,把 C1～C4 送入 PPU 的＄3F00～＄3F0F,具体地址由 M 值决定;若＄65≠0 则令＄63=7,把 C1～C4 送入 PPU 的＄3F10～＄3F1F,具体地址由 N 值决定。

M、N 的取值与 PPU 配色区的对应地址见表 4—8。

表 4—8 配色代码与 PPU 配色单元地址的对应关系表

背景配色代码 M	PPU 配色地址	卡通配色代码 N	PPU 配色地址
0	＄3F00～＄3F03	0	＄3F10～＄3F13
1	＄3F04～＄3F07	1	＄5F14～＄3F17
2	＄3F07～＄3F0B	2	＄3F18～＄3F1B
3	＄3F0C～＄3F0F	3	＄3F1C～＄3F1F

系统设定的配色面板代码默认值为"CGSET 1,1",各配色代码对应的颜色代码组合与 PPU 的对应关系见表 4—9。

表 4—9 系统设定的配色代码的颜色组合数据表

配色面板种类	面板编号	配色代码	颜色代码				存入PPU的地址
			C1	C2	C3	C4	
背景页	00	0	0F	2C	15	07	$3F00～$3F03
		1	0F	27	21	12	$3F04～$3F07
		2	0F	29	36	17	$3F08～$3F0B
		3	0F	30	26	07	$3F0C～$3F0F
	01	0	0F	30	21	02	$3F00～$3F03
		1	0F	30	27	18	$3F04～$3F07
		2	0F	30	27	16	$3F08～$3F0B
		3	0F	29	36	17	$3F0C～$3F0F
卡通页	00	0	0F	36	16	02	$3F10～$3F13
		1	0F	27	30	19	$3F14～$3F17
		2	0F	35	25	17	$3F18～$3F1B
		3	0F	30	27	16	$3F1C～$3F1F
	01	0	0F	30	16	01	$3F10～$3F13
		1	0F	10	00	01	$3F14～$3F17
		2	0F	30	29	09	$3F18～$3F1B
		3	0F	30	16	07	$3F1C～$3F1F
	02	0	0F	30	26	12	$3F10～$3F13
		1	0F	30	15	12	$3F14～$3F17
		2	0F	30	12	16	$3F18～$3F1B
		3	0F	30	26	19	$3F1C～$3F1F

4·6·3 背景画面的染色

背景画面的染色指令为 COLOR。语句格式为

COLOR X,Y,N

功能是对 F BASIC 背景画面座标为(X,Y)的显示单元染 N 号配色代码的颜色。COLOR 指令的解释程序入口在 $BA4E。执行过程如下：

1. 调 $9C6D 读出 X 座标值存入 $48 单元；调 $9C6A 读出 Y 座标值存入 $49 单元；调 $9C6A 读出配色代码 N 存入 $9C 单元。

2. 调 $B9EC 令 $48+2、$49+3，根据 $48、$49 单元的数值计算出对应的 PPU 座标存入 $4A、$4B；然后根据 $49、$48 的取值对 $19、$1A 赋值，令 $63=5 读出 $4A、$4B 指向地址的原配色数据存入 $73 单元。

3. 令(X,Y)座标单元的配色与 $1A 进行"与"运算后，再与 $9C 进行"或"运算，结果存入 $73 单元；然后令 $63=9，把新的配色数据置入(X,Y)点的 PPU 配色单元。

由于 COLOR 指令的执行过程较难理解，下面对照图 4—2 分析它的执行过程。

图 4—2 是 F BASIC 显示画面座标与配色地址的对应关系图。图中标出了 COLOR 指令中的 X、Y 座标的十进制、十六进制数值，方框中是与该显示单元对应的配色地址。

现以 COLOR 9,4,2 为例，并假定 $23CA=HE4，讨论 COLOR 指令的执行过程。

(1) 调 $B9EC 令 $48+2=H0B、$49+3=07、$9C=2，然后令 A= $49=7 与 HFC 进行"与"运算后左移一位送入 $4A，有 $4A=8。

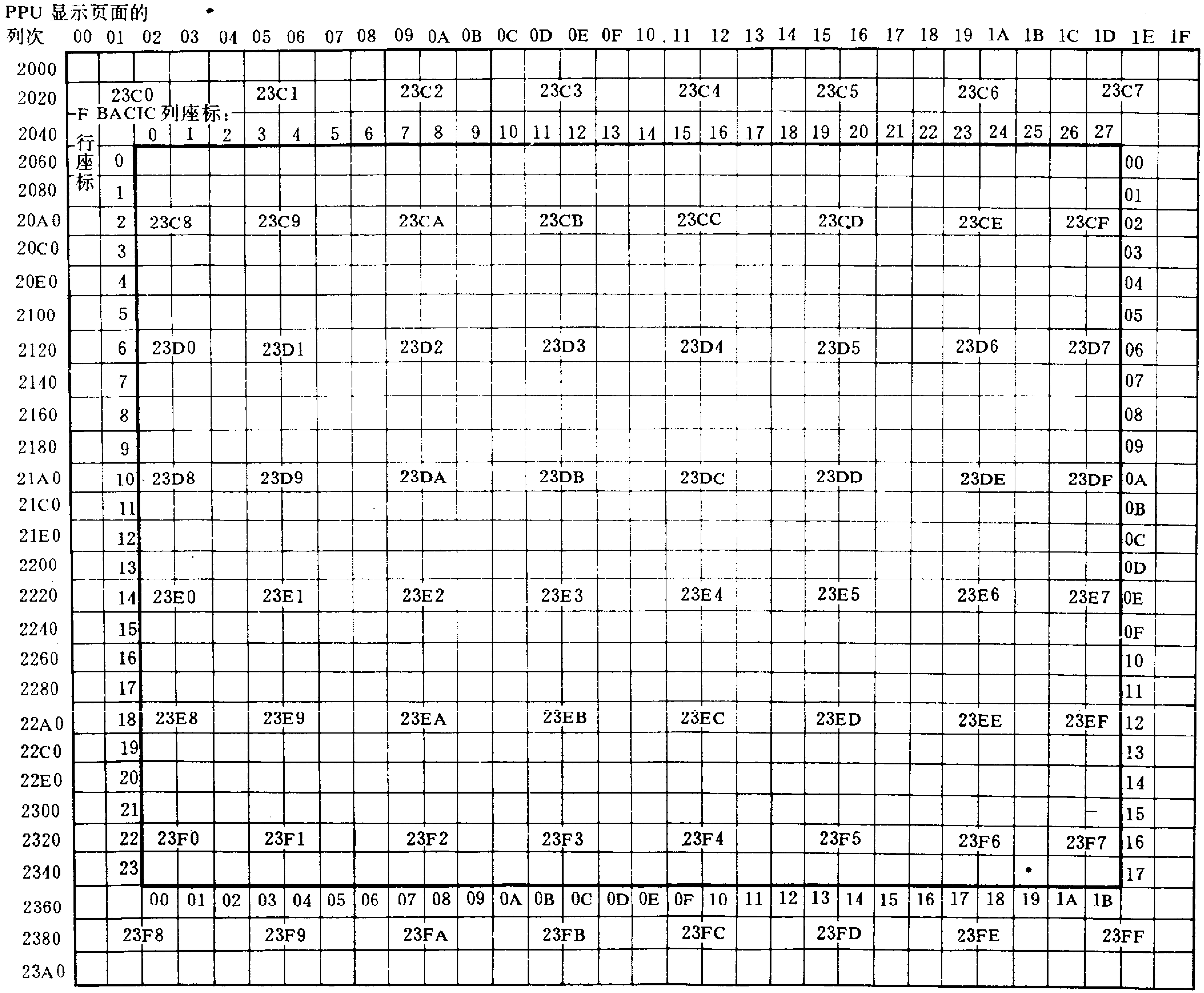


图 4－2　F BASIC 画面与配色单元的对应关系

(2) 令 A＝＄48＝H0B、右移两次后与＄4A 相加，再加 HC0，送入＄4A。现在＄4A＝HCA。令＄4B＝＄4F＋3＝H23。

通过以上计算后，座标(9，4)的配色地址为＄23CA，分别存在＄4A、＄4B 单元，见图 4－2 的五星标记。

由于＄23CA 存储了 16 个显示单元的配色，而 COLOR4，9，2 的目的是使＄23CA 配色大方格中的右下小方格的四个单元染为 2 号配色，其它三个小方格的配色不变。因此，进行以下处理。

(3) 令＄29＝2 AND ＄49＝2；

令＄48 AND 3 后右移一位与＄19 单元相加后存入＄19，有＄19＝3；

令＄1A＝HFC、＄19＝＄19×2＝6。

(4) 因＄19≠0，故令＄9C 左移位六次，＄9C＝H80；

令＄1A 循环右移六次，＄1A＝H3F。

(5) 令＄63＝5，取出＄23CA 的配色数据存入＄73。按假定有＄73＝HE4。

(6) 返回＄BA67 后，令

＄73 AND ＄1A OR ＄9C＝HA4

送入＄73，然后令＄63＝9，把配色数据送入＄23CA。

COLOR 4，9，2 执行前后＄23CA 的变化对比如下：

00	01
10	11

执行 COLOR4，9，2 之前
＄23CA＝HE4

00	01
10	10

执行 COLOR4，9，2 之后
＄23CA＝HA4

4·6·4 BEEP 发声过程

BEEP 指令的功能是发出蜂鸣声，解释程序入口在＄B30F。

执行＄B30F：首先令＄19＝F0、＄4015＝1、＄4017＝C0、＄4000＝EF、＄4001＝0、＄4002＝5F、＄4003＝0，使第一主声部发声；然后用＄19 单元为计数器控制发声延时时间，当＄19＝0 时，延时结束，令＄4015＝0 关闭发声。

执行＄B30F 可控制第一主声部发声延迟约一秒钟。

通过 BEEP 的分析可见，当执行＄B314 时令＄19＝1，从而可控制一声部发出不到 0.1 秒钟的“噗”声(即按键声)。因此，在 F BASIC 的 V3.10 版本中专门设置了 CLICK ON、CLICK OFF 指令用于关闭和打开按键时的发声。

4·6·5 音乐旋律的演奏

PLAY 指令为音乐旋律演奏的专用指令。语句格式为

PLAY “A 系统：B 系统：C 系统”

功能是令三个音乐系统进行三重音演奏。也可以只用一个系统演奏。PLAY 指令的解释程序入口在＄9A1D。

执行＄9A1D：首先调＄9C7C 读数子程序，把 PLAY 语句引号中的字符串读出存入＄780

及以后的内存单元中，并由＄400记录串长、＄401和＄402指向＄780，由＄21单元记录七页的自由区首址。

第二、把＄401、＄402中的字串首址送入＄09、＄0A单元，并对＄91、＄92单元置入地址指针＄700。调＄9C40把存于＄780以后的字符串转移到＄700以后的单元中。存入方式如下：

A系统存入＄700～＄720单元，＄C6存其在七页的首址，置入00；

B系统存入＄721～＄741单元，＄C7存其在七页的首址，置入H21；

C系统存入＄742～＄762单元，＄C8存其在七页的首址，置入H42。

对每个系统字符后的一个单元置入零以作为结束标志。

第三、令＄C9＝7，置演奏标志；令＄CB＝＄CC＝＄CD＝1，置三系统有效标志(当某系统无弦乐资料时则对应单元置零)；令＄4001＝H7F，＄4005＝H7F、＄4015＝07，＄63＝H0A，开始演奏弦乐资料。同时扫描STOP键，若按STOP键则停止演奏。

第四、检查＄C9，若＄C9≠0则继续演奏；若＄C9＝0表示结束演奏，进行结束处理。

各音乐系统使用的有关RAM单元示于表4－10。

表4－10 PLAY语句中各系统使用的RAM单元汇总表

	A系统	B系统	C系统
弦乐资料存于七页的内存地址 (资料后的第一单元置零)	＄700～＄720	＄721～＄741	＄742～＄762
首址记录单元	＄C6＝0	＄C7＝21	＄C8＝42
演奏有效标志单元	＄CB＝1	＄CC＝1	＄CD＝1
演奏无效标志单元	＄CB＝0	＄CC＝0	＄CD＝0

PLAY指令的执行共分两部分：以上讨论的是第一部分，作用是把语句中的弦乐资料置入七页，对演奏有关单元置值；第二部分将在中断程序中完成。关于中断处理部分我们将专题讨论。

小　结

· CGSET指令的执行实质是：把由M、N指定的配色组合数据送入PPU的配色单元。配色数据位于＄C3AF～＄C3FE。

· PALET指令的执行实质是：把用户指定的C1～C4颜色代码送入指定配色代码的PPU配色单元。

· COLOR指令的执行实质是：读出指定座标对应的配色数据，屏蔽对应位后，再把指定的配色代码送入对应位，形成新的配色数据存入对应的配色单元。

· BEEP指令的执行实质是：令第一主声部发出约一秒钟的蜂鸣声。

· PLAY指令的执行实质是：把各系统的弦乐资料送入＄700～＄762单元，并对＄C6～＄CD演奏工作单元置值，对＄4001、＄4005、＄4015发声寄存器置值后，令＄63＝A就可演奏。演奏过程由CPU的中断程序处理。

4·7　键盘输入指令和功能键指令的讨论

键盘输入指令有三条：单键输入指令INKEY＄、多字符输入指令INPUT、LINPUT。功能键指令有两条：功能键定义指令KEY、功能键定义列表指令KEYLIST

4·7·1 INKEY＄ 指令的讨论

INKEY＄指令的功能是读出任一按键的代码内容，它属于立即读键型指令（不须按回车键）。解释程序入口地址在＄A520。

执行＄A520：首先调＄8378读数子程序，判断INKEY＄后是否跟有参数。若有参数则调＄A2C3读出括号中的参数存入＄28、＄29单元；然后调＄B263读键盘，直到按任一键后转＄A466，把所读键码送入七页，并令＄28＝1，＄29、＄2A记录键码存于七页的地址；最后把键码送入四页以便存入变量区或送显。

若INKEY＄后没有参数则转＄A532，首先调＄AF88读键盘，若没有按键则令＄28＝＄29＝＄2A＝0，结束INKEY＄处理；若有按键则调＄A473把键码存入七页，令＄28＝1，＄29和＄2A记录存入七页的地址；然后把＄28～＄2A的内容送入四页以备存入变量区或送显；最后令＄27＝3，结束INKEY＄处理。

4·7·2 INPUT 指令的讨论

INPUT指令的功能是从键盘上输入数据资料，属于暂缓读键型指令（按回车键后有效）。语句形式为

INPUT"提示信息"；A1，A2，A3………

或　INPUT"提示信息"，A1，A2，A3………

其解释程序入口在＄93C2。

1. 若INPUT后有引号则调＄94DB读出左引号后的字符送入七页，令＄63＝1，显示提示信息，一直读到右引号时结束送显，进行提示信息后符号的判断处理。

2. 提示信息后有两种符号供选择：若为分号则调＄AC0D送显问号；若为逗号则不显问号。

3. 调＄B343读取操作者输入的内容，每读一个送显一个，直到按回车键后把输入的内容送入五页，由＄52单元记录存于五页的首址。

4. 调＄9F75判断变量类型，建立变量区；然后把＄09、＄0A的程序地址指针送入＄9A、＄9B单元保存，把＄52的内容送入＄09、令＄0A＝5，使＄09、＄0A指向五页输入字符的首址。

5. 若是数值变量则调＄A4FE，把输入的数据转换为十六进制数存入变量区；若是串变量则转＄945B把串值从五页读出送入七页的＄700以后的单元，并由＄1B记录字串长度；然后把串长存入＄400，令＄401和＄402指向＄700，调＄82EF把串值从七页取出存入串变量数据区。

6. 从＄9A、＄9B取回程序地址指针存入＄09、＄0A中，读取以后的程序内容。若读出的内容是H2C则继续读键盘；若不是H2C则结束INPUT处理。

4·7·3 LINPUT 指令的讨论

LINPUT指令的功能是从键盘读取一整行（最多31个字符）字符，而且连提示信息也作为资料一同输入。其解释程序入口在＄94F6。

1. 判断LINPUT指令后面是否有引号。若有引号则调＄94DB送显提示内容。

2. 调＄B343读取从键盘输入的字符，直到按回车键后把输入的内容送入五页。

3. 调＄9F75 读取变量名，建立变量区。

4. 执行＄951B，把存于五页的内容取出送往七页。

5. 执行＄953A，令＄400 存串长，令＄401、＄402 指向＄500。转＄82EF 把输入的字符串送入变量区，并检查字符个数，若超过 31 个则进行出错处理。

4·7·4 功能键指令的讨论

KEY 指令的语句形式为

KEY N,"字符串"

功能是定义"N"号功能键代表字符串内容。KEYLIST 的功能是列出各功能键的定义内容。KEY 保留字的解释程序入口在＄95AA。

执行过程如下：

1. 取出 KEY 以后的数据，判断其是否为 H86(LIST)。

2. 若不是 H86 则调＄9C6D，读出功能键"N"的代码入栈；调＄9C7C，读出功能键的字符串存于＄780 以后的单元中，令＄400 存串长，＄401、＄402 指向串首址；N 值出栈减一后左移四次送入 X 寄存器，以 X 变址从七页取出功能键定义的内容存入＄480 以后对应的功能键定义单元中，并在最后置一个零作为定义结束标志。

3. 若是 H86 则转＄95F3，以＄90 单元为功能键号码的 ASCII 码计数器，以＄74 单元为功能键码计数器，首先向七页送入"F"的 ASCII 码 H46，然后送入功能键号码，紧接着送入一个空格 H20。再令＄74 的内容左移四次送入 Y 寄存器，以 Y 变址，从＄480 开始取出功能键的定义字符送入七页。若读出的数据小于 H20 时则由七页送入 H28、H4D、H29(以便送显"(M)")；若读出的数据为零则表示本功能键定义内容读数结束，故转＄963C，调＄87FB 送显功能键定义。

4. 当一个功能键定义内容显示完毕后，令光标下移一行，令＄74 加一指向下一个功能键。

5. 检查＄74 单元，若＄74≤8 则转＄95FE 继续显示功能键定义；若＄74＞8 则表示功能键定义显示完毕，结束处理。

小　结

· INKEY＄指令的执行实质是：在读键程序中反复运行，直到按任一键后转出。

· INPUT 指令的执行实质是：送显用户输入的每一个字符，直到按回车键后，才判断输入的内容，合法后送入变量区。

· LINPUT 指令的执行实质是：连同提示信息、标点符号一齐作为数据资料送入变量区。

· KEY 指令的执行实质是：把指定功能键的定义送入四页对应的功能键定义区。

· KEYLIST 指令的执行实质是：列出＄480～＄4FF 中八个功能键的定义内容。

4·8 系统操作指令的讨论

BS. 2A 版本的 F BASIC 操作系统中与系统操作有关的指令有 12 条，分别为：

SYSTEM	CLEAR
NEW	RUN
LIST	PAUSE
FRE	POKE

PEEK　　　　STOP

CONT　　　　END

下面分别讨论。

4·8·1　SYSTEM 指令的讨论

SYSTEM 指令的功能是返回 F BASIC 选项画面。解释程序入口在＄80D7。

执行＄80D7：首先令＄703A＝H5A、＄703B＝H33、＄00＝H33、＄703C＝＄07、＄703D＝＄08；然后转＄80AD 绘制选项画面，等待操作者按键。

4·8·2　CLEAR 指令的讨论

CLEAR 指令的功能有两个：当 CLEAR 后带参数时，功能是规定用户使用的内存空间；当 CLEAR 后没有参数时则清变量区。其解释程序入口在＄9226。

执行＄9226：首先检查 CLEAR 后面是否带有参数，若没有则进行清变量区的处理：

第一、令程序区末址加一得出变量区首址存入＄1D、＄1E 单元；

第二、令变量区首址加一得出变量区末址存入＄1F、＄20 单元；

第三、向变量区连续置入两个零作为空区标志；

第四、转＄98FA，令程序区首址加三存入程序体首址单元＄B2、＄B3，再令＄B4＝0，结束处理。

若 CLEAR 后面有参数时则转＄924D 进行设定用户 RAM 区处理：

第一、调＄9228，执行 CLEAR 清除变量区；

第二、取出 CLEAR 后的参数存入＄400、＄401 单元；

第三、把参数存入自由区末址存储单元＄03、＄04。

4·8·3　清除程序区

NEW 指令的作用是清除用户程序。其解释程序入口在＄8382。

执行＄8382：首先令＄24＝0 置 CONT 指令无效；然后令程序区首址开始的连续三个单元为零，即＄703E＝＄703F＝＄7040＝0；最后向＄05、＄06 单元置入程序区首址＄703E，＄07、＄08 单元置入程序区末址，＄1D、＄1E 单元置入变量区首址＄7040，＄1F、＄20 单元置入变量区末址＄7041，＄21、＄22 单元置入自由区首址＄7042。再转 END 指令处理，进入读键等待状态。

4·8·4　暂缓执行型程序的运行

RUN 指令功能是运行存入程序区的暂缓执行型程序。RUN 的解释程序入口在＄857C。

1. 当键入 RUN 指令回车后，经翻译有＄300＝82、＄301＝0，执行＄8275，首先扫描 STOP 键；然后检查堆栈情况；最后由＄09、＄0A 单元指向＄300，取出 RUN 的保留字代码 H82，经保留字转向处理程序＄8321～＄833F 处理后转向 RUN 保留字的解释程序入口＄857C。

2. 执行＄857C：首先调＄9228 清变量区；然后调＄A96E 对六页单元初始化；第三、调＄A6DF 使二页单元初始化以解除卡通定义；第四、调＄8378，检查 RUN 后面是否有参数（行号）：若有参数则转＄9146 进行 GOTO 语句的处理，从指定行号开始运行程序；若无参数则从程序区首址＄703E 开始执行程序，把程序区首址存入＄09、＄0A 后转＄834D；第五、取第一

行程序占用 RAM 单元数，计算下一程序行首址存入＄0D、＄0E 单元，读出第一行行号存入＄0B、＄0C 后，转＄8275 读取程序体内容。第六、对程序体内容进行判断：若是保留字则转＄82DD 以控制转入相应的保留字入口；若是大写字母则调＄9F75 建立变量区，检查变量后的内容，调＄9C82 表达式取数处理子程序，计算出表达式的值送入＄400 单元暂存，最后送入变量区。

3. 返回＄8275，执行后续程序。

4·8·5 暂缓执行型程序的列表

LIST 指令的功能是列出存于程序区的程序清单，其解释程序入口在＄85AF。执行过程如下：

1. 令列表起始行号存储单元＄86＝0、＄87＝0，令列表结束行号存储单元＄88＝HFF、＄89＝HFF。即置 0～65535 行全部列表。

2. 检查 LIST 指令后的内容：

若为“LIST” 则转＄8604，列出内存中所有程序；

若为“LIST ，”或“LIST －”则转＄85E5，进行列表结束行号的读入处理，然后转列表处理；

若为“LIST N”则读出 N 值存入＄86～＄89，即令列表起始行号和结束行号为同一值。

3. 检查“LIST N”后的内容：

若为“LIST N”则转＄8604 只列出行号为 N 的程序行；

若为“LIST N ，：”或“LIST N －：”则转＄85FE 进行从指定行号 N 开始列表到行号为 65535 为止的处理；

若为“LIST N ，M”或“LIST N－M”则读出 M 值存入终止行号存储单元＄88、＄89 中。

4. 把列表起始行号送入＄44、＄45 工作单元，调＄8404 子程序，从程序区首址开始检索欲取行号的首址，检索到后把它存入＄5B、＄5C 单元。

5. 取出当前程序行行号存入＄28、＄29，取出列表结束行行号存入＄2C、＄2D，调＄8BF7 比较判断是否达到结束行号：若大于结束行号则结束列表；若小于或等于结束行号则继续列表。

6. 读出行号，转换为 ASCII 码后送入七页，然后调＄900A 读取程序体，进行判断处理。

(1) 从程序体读出的数据若小于 H20 则表示为十进制数或十六进制数，转＄902E 把数据转换为 ASCII 码送入五页。若是十六进制数据，还要把“&H”先送入五页。

(2) 从程序体读出的数据若大于或等于 H20、小于 H80 则表示该数据是字母或字符，于是转＄9064 进行判断处理：

若是双引号则把引号中的内容读出送入五页；

若是单引号则把单引号后的全部内容读出送入五页(直到本行行尾)；

若是变量则把变量送入五页。

(3) 从程序体读出的数据若大于 H80 则表明是保留字，于是调保留字检索程序＄9094，检索到保留字代码后，把保留字字符送入五页。当本行程序全部反编译完毕送入五页后，再置入一个零表示本行程序反编译结束。

(4) 令＄52、＄53 指向＄500，把五页的内容送往七页显示。

7. 扫描键盘：若按“STOP”键则结束列表；若按“ESC”键则暂停列表，松开“ESC”键时再继

续列表。

8. 重复五～七项，直到满足列表要求后结束列表处理。

4·8·6 程序运行中的暂停

PAUSE 指令的功能是令程序暂停执行，其解释程序入口在＄9198。

执行＄9198：若 PAUSE 后没有参数则转＄B263 读键程序等待按键，直到按任一键后才继续执行程序；

若 PAUSE 后有参数，首先检查参数值的大小：若大于 32767 则转出错处理；若为零则直接返回；若为 PAUSE N(N ≠0)则把 N 值存入＄400、＄401，并令＄401 加一，以＄74 为第一延时器、＄400 为第二延时器、＄401 为第三延时器，反复扫描“STOP”键，每扫描一次令＄74 减一，当＄74=0 后再令＄74－1、＄400－1，当＄400=0 后则令＄74－1、＄400－1、＄401－1，直到＄401=0、＄400=0、＄74=0 则暂停结束，继续执行后续程序。

4·8·7 RAM 空区的测试

FRE 指令的功能是测试用户 RAM 区的自由单元数量，其解释程序入口在＄A31A。

测试过程是：首先调＄A320，从＄03、＄04 单元取出自由区末址，减去＄1F、＄20 中的变量区末址，计算出 RAM 空区的数据存入＄28、＄29；然后转＄A2FB，把＄28、＄29 单元的内容送入＄400、＄401 以备存入变量区或送显。

4·8·8 POKE 指令的讨论

POKE 指令的语句形式为

POKE N，M1……

功能是向指定内存地址 N 写入数据 M1。其解释程序入口在＄91DF。

执行过程如下：

第一、调＄9C82，读出写入内存地址 N 存入＄400、＄401 单元，再转存于＄25、＄26 单元。

第二、调＄9C6A，读出写入数据 M1 存入 A 累加器。

第三、将 A 累加器的数据送往＄25、＄26 单元指向地址的内存 RAM 中。

第四、检查 POKE N，M1 后是否还有数据 M2、M3……，若有则令写入地址加一，把 M2 写入 RAM；若没有则结束写入操作。

4·8·9 PEEK 指令的讨论

PEEK 指令的语句形式为

PEEK（N）

功能是读出指定内存单元 N 中的数据。解释程序入口在＄A358。执行过程如下：

1. 调＄A3E3，读出括号中的内容存入＄28、＄29 单元。
2. 把＄28、＄29 指向内存单元中的数据读出存入＄28 单元，并令＄29=0。
3. 转＄A301 把＄28、＄29 的内容送入＄400、＄401 单元，以备送入变量区或送显。

4·8·10 运行程序的中断

STOP 指令的功能是中断程序的运行，返回按键等待状态。其解释程序入口在＄86DA。执

行过程如下：

1. 出栈两次，令 $ 63＝0，关闭系统通信开关。

2. 调 $ B3F1，令 $ 30 OR H80 → $ 2000。

3. 把当前执行的程序行号存入 $ 11、$ 12 单元，把下一程序行首址存入 $ 13、$ 14 单元，把当前处理的指令地址存入 $ 15、$ 16 单元。

4. 令 $ 24＝HFF，置 CONT 指令有效标志。然后检查当前行号，若行号为 65535 则置 CONT 指令无效（$ 24＝0）。

5. 转 $ A9EA，显示按 STOP 键后的提示“BREAK”。

6. 转 $ A9B8，把字母“IN”的 ASCII 码送入七页。并把 $ 0B、$ 0C 中的中断行号转换成 ASCII 码送入七页显示。

7. 发出蜂鸣声，转 $ 8221 令光标下移一行，送显“OK”，置 $ 24＝0，令光标下移一行，进入读键等待状态。

4·8·11 CONT 指令的讨论

CONT 指令的功用是继续执行按 STOP 键后中断的程序（从原断点开始执行），解释程序入口在 $ 8717。执行 $ 8717：

第一、检查 $ 24 单元，若 $ 24＝0 表示 CONT 指令无效，转出错处理。

第二、把中断前指令的地址由 $ 15、$ 16 取回送入 $ 09、$ 0A，调 $ 9908 取数判断：若是保留字则令 $ 09＋1 后返回；若是十进制数、十六进制数标志则令 $ 09＋3 后返回；若是引号则转 $ 9934 跳过引号中的内容，令 $ 09、$ 0A 指向引号后的第一个数据；若是其它数据则调 $ A1C3 进行第二层判断：若是大写字母则转 $ 9942，否则令 $ 09＋1 返回；执行 $ 9942 读数，若大写字母后仍是大写字母或数字则令 $ 09＋1 再读，直到不是字母、数字后返回。即跳过变量名，使 $ 09、$ 0A 指向变量名后的单元。

第三、调 $ 8378 判断读出的数据是否为 H3A，若不是 H3A 则令 $ 09＋1 后再读，直到读数是 H3A 后执行 H3A 以后的程序。

第四、执行 $ 872B 把下一程序行的首址从 $ 13、$ 14 中取出存入 $ 0D、$ 0E，把原执行程序行的行号从 $ 11、$ 12 单元取出存入 $ 0B、$ 0C 单元，令 $ 24＝0，结束 CONT 处理。

4·8·12 END 指令的讨论

END 指令的功能是结束程序的运行，其解释程序入口在 $ 820B。

执行 $ 820B：首先令 $ 24＝0、$ 703A＝0、$ 703B＝0，设置堆栈顶为 HFF，对栈顶置入两个 HFF，调 $ ADA9 置方块光标，把栈指针存入 $ 17 单元。

然后令光标下移一行送显“OK”，再令光标下移一行进入读键状态。

小　　结

1. SYSTEM 指令的执行实质是对 $ 703A～$ 703D 赋值后绘制 F BASIC 选项画面，等待操作者选项。

2. CLEAR 指令的执行实质是清除变量区；当后面跟有参数时则把参数存入 RAM 自由区末址存储单元 $ 03、$ 04 中。

3. NEW 指令的执行实质是向程序区末址单元置入 $ 703F，变量区首址单元置入 $ 7040，变量区末址单元置入 $ 7041，自由区首址单元置入 $ 7042。

4. RUN 指令的执行实质是清变量区、清六页、清二页，然后读出程序体数据进行判断处理。

5. LIST 指令的执行实质是根据 LIST 后的参数分别对起始行号存储单元＄86、＄87 和结束行号存储单元＄88、＄89 赋值，然后根据＄86～＄89 的数据对程序列表。

6. PAUSE 指令的执行实质是根据设定的参数向延时控制器＄74、＄400、＄401 单元赋值，当以上三单元计数递减至零时，延时结束继续执行后续程序。

7. FRE 指令的执行实质是取出 RAM 自由区末址减去变量区末址，把所得之差送入＄400、＄401 单元。

8. POKE 指令的执行实质是把 POKE 语句中指定的数据送入指定的内存单元。

9. PEEK 指令的执行实质是把指定单元的内存读出存入＄400、＄401 单元。

10. STOP 指令的执行实质是关闭系统通讯开关，保存断点现场，送显“BREAK IN ××××”，发出蜂鸣声，送显“OK”后进入读键等待状态。

11. CONT 指令的执行实质是取回断点现场，从原断点处继续执行程序。

12. END 指令的执行实质是置 CONT 指令无效，令＄703A＝0、＄703B＝0，清堆栈区，送显“OK”，令光标下移一行进入读键等待状态。

4・9　读操纵器指令和数值函数的讨论

读操纵器指令有两条：一条是读操纵器方向键指令 STICK、一条是读操纵器功能键指令 STRIG。数值函数有三个：为求绝对值函数 ABS、求数值符号函数 SGN、随机函数 RND

4・9・1　读操纵器的方向键码

STICK 指令的功能是读操纵器方向键码，其解释程序入口在＄A6AA。

执行＄A6AA：

1. 调＄A677 读出括号中的操纵器编号，然后读输入输出寄存器。若是 0＃操纵器则读＄4016；是 1＃操纵器则读＄4017。每读一次令 A 寄存器的内容右移一位，令＄90 单元循环左移一位，即由＄90 记录读操纵器的数值，连读八次后返回。

2. 调＄A67A 继续读＄4016(或 ＄4017)，每读一次令 A 寄存器的内容右移一位，令＄90 单元的内容左移一位，连读八次后返回。

3. 取＄90 单元的数值，屏蔽高四位后存入＄28 单元；再以＄30 变址把＄28 中的操纵器方向键码存入＄400＋＄30 单元，并令＄401＋＄30＝0、＄402＋＄30＝0，以备送入变量区。

4. 各方向键码如表 4－11。

表 4－11　STICK 指令中各方向键码汇总表

不动	向右	向左	向下	右下	左下	向上	右上	左上
0	1	2	4	5	6	8	9	10

4・9・2　读操纵器的功能键

操纵器上有四个功能键，分别为：STICK、SELECT、A、B，它们的键码由 STRIG 指令读出。STRIG 指令的解释程序入口在＄A6B9。

执行 ＄A6B9：

1. 调＄A677 读出括号中的操纵器编号，然后读输入输出寄存器。若是 0＃操纵器则读＄4016；是 1＃操纵器则读＄4017。每读一次令 A 寄存器的内容右移一位，令＄90 单元循环左移一位，即由＄90 记录读操纵器的数值，连读八次后返回。

2. 令＄90 单元的数值单步右移位四次后存入＄28 单元。

3. 以＄30 变址，把存于＄28 中的操纵器功能键码存入＄400＋＄30 单元，并令＄401＋＄30＝0、＄402＋＄30＝0，以备送入变量区。

4. 各功能键码如表 4—12。

表 4—12 STRIG 指令中各功能键码汇总表

按　　键	键　码	按　　键	键　码	按　　键	键　码
不　按	0	START＋B	5	SELECT＋A	10
START	1	SELECT＋B	6	A＋B	12
SELECT	2	A	8	START ＋SELECT	3
B	4	START＋A	9		

4·9·3　求数值的绝对值

ABS 指令的功能是对指定数值求绝对值，其解释程序入口在＄A369。执行过程如下：

1. 调＄A3EC 子程序读出括号中的数值存入＄28、＄29 单元。

2. 判断＄29 单元数值的大小。若＄29＞H7F 则表示数值为负数(因在＄A3EC 子程序读数中，若读到负号则对数值求补码存入 28、＄29 中)，于是令 C＝1、0－＄28→＄28、0－＄29→＄29，恢复原数值的绝对值；若＄29＜80 则不作处理。

3. 把＄28、＄29 的数据送入四页以备送入变量区或送显。

4·9·4　求数的符号

SGN 指令的功能是求出指定数的符号：若为正数其值为 1；若为负数其值为－1；若为零其函数值也为零。SGN 指令的解释程序入口在＄A379。执行过程如下：

1. 调＄A3EC 子程序读出括号中的数据存入＄28、＄29 单元。

2. 判断＄29 单元数值的大小。若＄29＞H7F 则表示为负数，于是令＄28＝HFF、＄29＝HFF，转＄A363 把＄28、＄29 的数值送入四页，以输出“－1”；

3. 若＄29＝0，再判断＄28 的大小，若＄28＝0 则转＄A363 把＄28、＄29 送入四页以输出“0”。

4. 若＄29＜H80 则表示为正数，令＄28＝1、＄29＝0，转＄A363 以输出“＋1”。

4·9·5　求随机数

RND 函数的功能是获取随机数，语句形式为

　　RND (N)　　　　N：取值为 1～32767

F BASIC 随机函数指令的特点是可以产生 0～(N－1)数据范围的随机数。其解释程序入口在＄A39F，执行过程如下：

1. 调＄A3EC 子程序读出括号中的随机数上限数值存入＄28、＄29 单元，返回后令＄28、＄29 的内容入栈。

2. 令＄28＝＄56、＄29＝＄57、＄2C＝H83、＄2D＝3，调＄8B59 乘法运算子程序，计算出

随机数基数存于＄28、＄29 单元。

3. 令＄29 AND H7F，结果存入＄29、＄57 单元，令＄56＝＄28；随机数上限出栈分别存入＄2C、＄2D；调＄8B59 进行乘法运算后，积存于＄28、＄29 中。

4. 令＄29 左移一位，令＄36、＄37 循环左移一位，得出随机数存于＄36、＄37 单元。

5. 令＄28＝＄36、＄29＝＄37，转＄A363 把存于＄28、＄29 中的随机数值送入四页以备输出。

小　结

1. STICK 指令的执行实质是：根据操纵器的编号决定读＄4016 或＄4017，读出的方向键码值存入四页以备送显。

2. STRIG 指令的执行实质是：根据操纵器的编号决定读＄4016 或＄4017，读出的功能键码值存入四页以备送显。

3. ABS 指令的执行实质是：去除数据的符号，由＄28、＄29 中转，把数据存入四页以备送显。

4. SGN 指令的执行实质是：根据数据为零、为正、为负，通过＄28 单元中转分别向四页送入 0、1、－1，以备输出。

5. RND 函数指令的执行实质是：根据随机数基数存储单元＄56、＄57 的内容和 RND 函数指令中规定的随机数上限，通过乘运算和移位操作产生随机数存于＄36、＄37 单元。最后通过＄28、＄29 中转送入四页以备输出。

4·10　循环语句指令的讨论

循环语句由四个保留字组成：FOR、TO、STEP、NEXT。循环语句的执行共分三个阶段：第一阶段执行 FOR ～ TO ～ STEP；第二阶段执行循环体；第三阶段执行 NEXT。

FOR 指令的解释程序入口在 ＄927E。执行过程如下：

1. 首先取出栈顶两单元内容存入＄5F、＄60；然后把下一行程序首址从＄0D、＄0E 中取出入栈，把循环程序行行号从＄0B、＄0C 中取出入栈。

2. 调＄8CD4，读出循环变量名存入＄8A、＄8B，变量类型存入＄90，变量值首址存入＄8C、＄8D，并令变量值首址入栈保护。

3. 调＄9C82 取出循环变量初值存入循环变量数值区，再调＄9C82 取出循环变量终值入栈。

4. 调＄9C82 取出循环步长值入栈，若没有 STEP 则令数值 1 入栈。

5. 循环体首址指针入栈；令“0”、“HFD”入栈作为循环语句的参数栈顶标志。

6. 转＄911C，把原栈顶两单元的内容从＄60、＄5F 中取出入栈(原栈顶标志)，结束 FOR 语句的处理。

7. 执行循环语句体语句。

NEXT 指令的解释程序入口在 ＄930D。

当执行完循环体语句，读到 NEXT 指令时则转＄930D。NEXT 指令执行过程如下：

1. 栈顶标志出栈，存入＄5F、＄60；循环参数栈顶标志出栈，若不是 HFD、0 则进行出错处理；循环体首址出栈，存入＄88、＄89 单元；步长出栈存入＄83、＄82 单元；循环终值出栈，存入＄81、＄80；循环变量值首址出栈，存入＄7D、＄7C；循环语句所在行号出栈，存入＄87、＄86

单元;循环语句的下一行程序首址出栈,存入 $85、$84 单元。

2. 取出循环变量当前值,调 $938A,令循环变量值加一个步长,判断是否大于循环终值:若大于终值则令原栈顶标志内容由 $60、$5F 单元取出入栈,退出循环语句的处理,返回调用程序。

3. 若循环变量值小于或等于终值时,则从 $84、$85 中取出下一行程序首址存入 $0D、$0E 单元,并入栈;从 $86、$87 中取出循环语句行号存入 $0B、$0C,并入栈;从 $7C、$7D 中取出循环变量值首址入栈;从 $80、$81 中取出循环变量终值入栈;从 $82、$83 中取出循环步长入栈;循环体首址送入 $09、$0A,并入栈;循环参数栈顶标志 0、HFD 入栈;原栈顶标志 $5F、$60 内容入栈。

4. 继续进行循环体语句的处理。

小　　结

执行循环语句的实质是:每执行一次循环体,令循环变量加一个循环步长;然后与循环终值比较,若小于或等于终值则继续执行循环体;若大于循环终值则结束循环,退出循环语句的处理。

4·11　其它指令的讨论

4·11·1　非执行指令的讨论

F BASIC 中有两条非执行指令:注释指令 REM 和数据说明指令 DATA。它们的解释程序入口在 $8343,执行过程如下:

1. 执行 $8343,出栈两次后,取下一行程序首址存入执行程序地址指针 $09、$0A。
2. 取下一行程序长度与 $09、$0A 相加后得出第三行程序首址存入 $0D、$0E 单元。
3. 取下一行程序行号存入 $0B、$0C 单元。
4. 令 $09 单元加 3,使程序地址指针指向下一行程序体,继续执行下一行程序。

4·11·2　读数指令 READ

READ 指令的功能是从程序区 DATA 语句中读出数型变量值或字串型变量值,其解释程序入口在 $994D。执行过程如下:

1. 调 $9F75 读取变量字符子程序,把变量名存入 $8A、$8B 单元;然后检索变量区,若检索到相同变量名则令 $8C、$8D 指向变量值首址;若没检索到相同变量则建立变量区,使 $8C、$8D 指向变量值存放首址。

2. 调 $8FF8 保存 $09、$0A 程序地址指针于 $9A、$9B 单元;然后从程序区第一行程序体开始检索 DATA 保留字代码 H91:若检索到则转 $997E 进行后续处理;若没检索到则打印“OD”(数据不足)出错提示。

3. 执行 $997E:首先令 $B4=HFF,置检索到 DATA 标志;然后判断所读数据类型:若是串型变量则转 $99CB 进行读字串处理;若是数型变量则调 $A4FE 把数值的 ASCII 码转换成十六进制数据存入 $28、$29 单元,再经 $28、$29 中转存入变量区。

4. 读完一个变量数据后,再检查判断 READ 语句中后面是否还有变量名。若是 0 或 H3A 则说明 READ 语句执行完毕,令 $B4=0,置读 DATA 结束标志;若是 H2C(,)则说明 READ

语句中还有待读变量，则令＄09＋1送入＄B2、＄0A送入＄B3，使＄B2、＄B3指向DATA语句中下一个数据的首址，然后读出下一个变量名，重复一～四的处理。

5. 当读串型变量时则执行＄99CB，由＄1B记录字串长度，读出字串存入＄700以后的单元；然后调＄82EF把字串送入串变量区；最后转＄99AA再读，直到本字串值全部读出后返回调用程序。进行第四步处理。

4·11·3 两变量值的交换

F BASIC有一条排序中极为有用的指令SWAP，它的功能是令两同类变量互相交换内容，其解释程序入口在＄9496。

执行 ＄9496：

1. 调＄9F75读出第一个变量值首址存入＄8C、＄8D，变量类型标志存入＄90单元；然后令三个单元的数值入栈保护。

2. 再调＄9F75读出第二个变量值首址存入＄8C、＄8D，变量类型标志存入＄90单元。

3. 令第一变量值首址出栈存入＄19、＄1A。然后比较两变量类型：若不同则转出错处理。（注——数型简单变量和数型下标变量同类。串变量亦然。）

4. 若两变量同类，再检查变量类别：若是数型变量则令X＝2；若是字串型变量则令X＝21（因数型变量占两个单元、字串型变量占H21个单元）。然后以X寄存器为计数器，以Y寄存器变址（初值为零），先从＄8D＄8C＋Y中取出第二变量的一个数据入栈；再从＄1A＄19＋Y中取出第一变量的数据送入＄8D＄8C＋Y单元；最后第二变量的数据出栈，存入＄1A＄19＋Y中，即两变量之间对应位逐个交换内容。每交换一个数据令X减1、Y加一，直到X等于零时结束交换。SWAP指令的执行过程如图4－3所示。

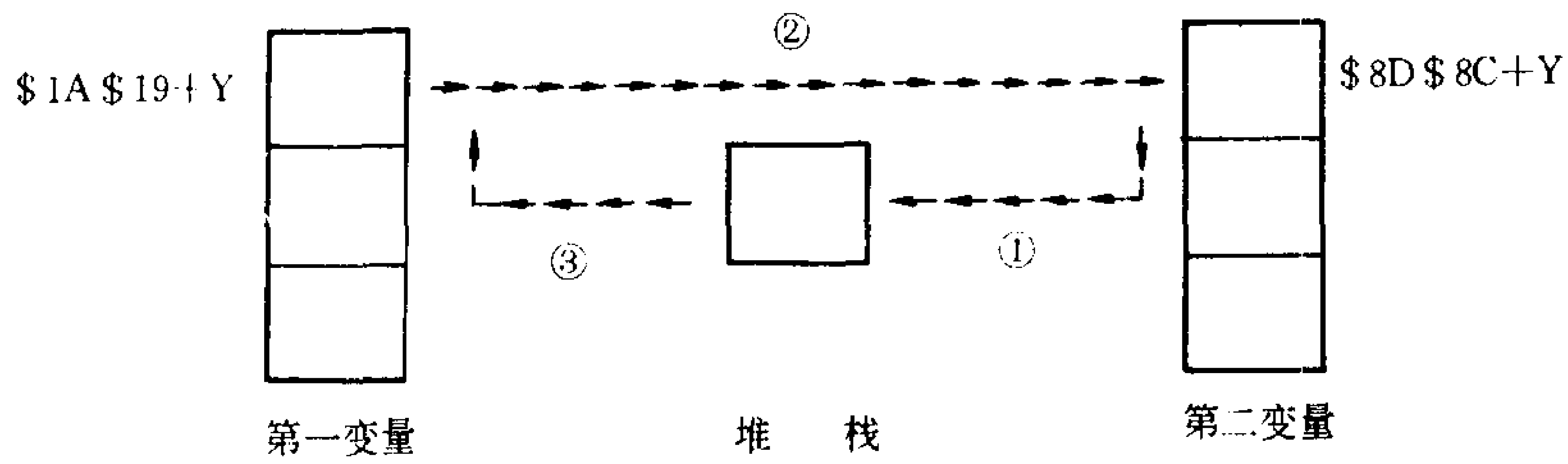


图4－3　SWAP指令执行示意图

4·11·4 数组说明指令的讨论

DIM指令的功能是说明数组，定义下标变量，其解释程序入口在＄964E。

定义下标变量的工作过程如下：

1. 调＄A1C3，判断DIM后的第一个字符，若不是大写字母则出错（H20除外。数组名的第一个字符必须是大写字母。）；然后把第一个字符存入＄7C单元。

2. 令＄84＝2，预置变量类型为数型下标变量；令＄7D＝0，预置下标变量名无第二字符。

3. 调＄8CAD读出下标变量名第一字符后的内容进行判断：若是数字0～9或大写字母A～Z则说明该数组名有两个字符，于是将第二个字符存入＄7D单元；若是“＄”则表明定义字串型数组，于是令＄84＋1，置串型变量标志；若是“（”则表示数组名只有一个字符；若不是以上

字符则进行出错处理。

4. 调 $ 9C64,读出第一下标值存入 $ 7E 单元;调 $ 8CD4 判断第一下标后的字符:若为“)”则表示定义的是一维数组;否则是两维数组,则令 $ 82+1,调 $ 9C6A 读出第二下标存入 $ 7F。

5. 调 $ 96B2 检索变量区,检查是否已定义过同名数组,若有则打印出错提示;若没有则建立本数组变量区,把本数组的有关标志写入数组变量区标志单元。

6. 判断 DIM 语句中是否还有欲定义的数组。若有则转 $ 964E 继续定义后面的数组;若没有则结束 DIM 的处理,返回调用程序。

4·11·5 外存取指令的讨论

F BASIC 的外存取设备为磁带录音机。外存取指令有三条:分别为写入磁带指令 SAVE、核对写入指令 LOAD? 或 LOADPRINT、从磁带取出指令 LOAD

一、SAVE 指令

SAVE 指令的功能是把指定的数据资料存入磁带中。解释程序入口在 $ 978C。

执行 $ 978C:

1. 调 $ 9869 检查 SAVE 指令后是否有文件名,若有文件名则读出存入 $ 580 开始的 RAM 中,令 $ B5=1,置有文件名标志。

2. 调 $ 98B4 把存入 $ 580 以后的文件名转存 $ 501～$ 510 单元,并令 $ 500=2、$ 511=0 。

3. 令 $ 514、$ 515 存程序区首址,$ 512、$ 513 存程序长度,调 $ 97C3 送显“WRITING”、送显文件名,令光标下移一行,位于行首。

4. 调 $ B4FC,首先令 $ 32 的内容送入 $ 2000 以关闭中断;然后反复向 $ 4016 写入 4、HFF,以产生写入磁带的高频导引脉冲信号;再从 $ 500 开始读出内存,并对 $ 4016 置入 4、HFF,使数据加载在脉冲信号上一同写入磁带;直到写入完毕,令 $ 32 OR H80 → $ 2000,开中断。

5. 转 $ B50D 计算程序区末址存入 $ AF、$ B0,令 $ 91=14、$ 92=0、$ 19=20、$ 1A=4E,再对 $ 4016 置入 4、HFF,发出写入结束信号,结束 SAVE 操作。

二、LOAD 与 LOAD? 指令

LOAD?指令的功能是核对写入磁带的数据是否准确无误。LOAD 指令的功能是从磁带中取回数据资料。LOAD 指令的解释程序入口在 $ 97FC。

执行 $ 97FC:

1. 检查 LOAD 后的内容:若语句形式为 LOAD 则令 $ 9F=0;若语句形式为 LOAD-PRINT 则令 $ 9F=HFF。

2. 检查有无文件名,若有文件名则把文件名读出存入 $ 580 开始的 RAM 单元,令 $ B5=1。

3. 调 $ B430,反复读 $ 4016,并扫描 STOP 按键。若按 STOP 键则结束操作,进行 STOP 处理。把读出的文件名存入 $ 501 开始的 RAM 单元。

4. 调 $ 98C8 核对读出的文件名是否与欲读的文件名相同。若相同则令 C=0;否则令 C=1。

5. 若文件名不同则送显“SKIP 文件名”,令光标下移一行,位于行首,转第三步重复 3～5

的操作。

6. 若文件名相同则送显“LOADING 文件名”。若语句为“LOAD”则把磁带中的数据读出装入内存;若语句为“LOADPRINT”则只检查数据资料是否正确。若正确则结束操作,若不正确则令 C=1 返回,送显“TP”出错标志,结束操作。

总之,外存取指令的操作是一个复杂的过程。我们不必全弄明白电脑为什么能够写入和读出,只要知道每一条指令的操作过程即可。

小　　结

1. 非执行指令 REM、DATA 的执行实质是:令程序地址指针 $ 09、$ 0A 指向下一行程序首址。

2. READ 指令的执行实质是:首先认定 READ 指令后的变量名;然后从 DATA 语句中读出数据存入认定变量的变量值存储单元。

3. SWAP 指令的执行实质是:以堆栈为媒介,两变量的对应内容相交换。

4. DIM 指令的执行实质是:按 F BASIC 程序的存储格式,在变量区开辟出下标变量的存储区和数组标志存储区。

5. SAVE 指令的执行实质是:根据 $ 514、$ 515 指出的保存程序首址和由 $ 512、$ 513 指出的保存程序长度,通过反复对 $ 4016 置入 4、HFF,把程序写入磁带中。

6. LOAD 指令的执行实质是:通过反复读 $ 4016 软开关,把欲取的数据资料从磁带中读出写入由 $ 514、$ 515 指定的 RAM 区。

7. LOADPRINT 指令的执行实质是:除不把磁带中的数据资料读入 RAM 以外,其它操作过程与 LOAD 相同。

4·12　F BASIC 中断处理程序的讨论

F BASIC 工作系统的中断处理主控程序在 $ 880F～$ 8A31,它主要完成三个方面的任务:

第一、完成与 PPU 的通信任务,其中包括所有屏幕编辑键的操作、PRINT、SCR $ 、VIEW、CLS 等与 PPU 有联系的指令处理。

第二、进行 PLAY 语句演奏处理。

第三、把定义动作卡通的数据从六页的有关单元取出,进行换算后送入卡通定义区,并完成向 PPU 的卡通数据传送;根据六页数据,控制卡通按 DEF MOVE、POSITION 语句的要求运动。

下面分别讨论它们的工作过程。

4·12·1　中断系列子程序

F BASIC 中断系列子程序的调用由 $ 63 单元控制,由 $ 882F～$ 8840 程序段根据 $ 63 单元的赋值控制中断处理转入不同的处理程序入口。中断入口数据区在 $ 8841～$ 885E。为便于查阅,特将各中断入口与 $ 63 单元的对应关系列于表 4－13。

表 4—13 中断系列子程序入口、功能汇总表

$63	入口地址	操 作 过 程	功 能
1	$88AA	计算出光标对应的 PPU 地址存入 $4B、$4A，然后置入 $2006，令 $700+X 的内容送入 $2007。	取出由 $64 指定的七页地址及以后的单元的内容送入 PPU，在屏幕上显示送显的内容
2	$8925	$48=0，计算光标的 PPU 地址存入 $4B、$4A，令 A=$2007→$500+X。	取出由 $49 单元指定的一行的内容存入 $500 及以后的单元。
3	$8961	以 $4B、$4A 为 PPU 地址，令 $2007→$500+X，然后以 $1A、$19 为 PPU 首址，令 $500+X→$2007。	取出由 $4B、$4A 开始的 PPU 中一行的内容存入由 $1A、$19 指定的 PPU 单元中。执行 DEL 时等。
4	$8898	$4B、$4A→$2006，X 计数，H20→$2007，$63=0	从光标开始清除到本行行尾。执行 Ctrl+Z。
5	$8870	$4B、$4A→$2006，$2007→$73，$63=0。	读出 $4B、$4A 指定的内容送入 $73 单元。执行 SCR$ 时等。
6	$8990	H3F→$2006，$700×4→$2006，$701～$704→$2007	把存于七页的背景页配色数据送入 PPU 的 $3F00～ $3F0F。执行 PALETB 时。
7	$89BE	H3F→$2006，$700×4 OR H10→$2006，$701～$704→$2007	把存于七页的卡通页配色数据送入 PPU 的 $3F10 ～ $3F1F。执行 PALETS 时。
8	$8882	$4B $4A→$2006，$73→$2007，$4B $4A→$2006、$61→$2007，$63=0。	读出 $4B、$4A 指定 PPU 的内容存入 $73，再把光标常态置入 $4B、$4A 指定的 PPU 单元中。
9	$8890	$4B、$4A→$2006，$73→$2007，$63=0	把 $73 中的数据送入由 $4B、$4A 指定的 PPU 单元，如执行 COLOR 语句等。
A	$9A90		PLAY 语句的处理(略)
B	$8862	$33→$2001，$63=0	开背景页显示。
C	$885F	0→$2001，$63=0	关背景页显示。
D	$89CC	由 $B0 中转依次把每行最后一个单元的内容转入下一行行首。	在光标处移出一个空格。执行 INS 时使用。
E	$8A02	$4B、$4A→$2006，X 计数，$2007→$500+X $，63=0F	读出由 $4B、$4A 开始的一行数据存入 $500 以后的单元，执行 INS 时使用。
F	$8A1A	$49=$49+1，$4B、$4A→$2006，$500+X→$2007	把存于五页的内容送入当前光标的下一行。执行 INS 时使用。

以上系列中断处理程序的结构较简单，不多赘述。

4·12·2 PLAY 语句的后期处理

我们知道，执行 PLAY 语句的操作，就是把各系统的弦乐资料按要求送入七页的有关单元，然后令 $63=A 就可演奏编写的音乐旋律。实际上保留字 PLAY 的解释程序仅仅完成了 PLAY 语句的赋值，大量的工作还是在中断处理中完成的。通过以上讨论，我们已了解到 $63=A 的中断处理程序入口在 $9A90，现在让我们看一下它的处理过程。

PLAY 语句的中断处理程序在 ROM 区 $9A90～$9C3F，其中 $9A90 为中断处理控制程序，$9AA2～$9AB7 为中断处理主控子程序，$9AB8～$9C0D 为中断处理工作子程序，

$9C0E～$9C3F 为音名处理子程序。下面我们仅以

PLAY "Y1M0V9M1V8T4O3C5R5D5"

为例分别讨论它们的作用。

以上语句的功能是：令 A 系统（第一主声部）以 Y1 音色、V9 音量、V8 音长、T4 节拍、发出 ‖ 1 0 2 ‖ 音符的音调。

由 PLAY 语句的讨论，我们知道 PLAY 保留字的解释程序将把以上 A 系统的弦乐资料送入七页的 $700～$713 单元，数据为

```
700  59  31  4D  30  56  39  4D  31  56  38  54  34
      Y   1   M   0   V   9   M   1   V   8   T   4
4F   33  43  35  52  33  44  35  00
 O    3   C   5   R   3   D   5
```

并令 $C6=0、$CB=1、$C9=7、$92=7、$63=A。

其中断处理过程如下：

1. 中断控制程序的处理

中断控制程序的处理过程是：保护 X、Y 寄存器的现场；令 $4017=HC0，作好发声准备；调 $9AA2 发声主控子程序，把弦乐资料的各参数置入有关单元；最后恢复 X、Y 的现场，中断返回。

2. 主控子程序的处理

. 令 $CA=2，以便从 C 系统开始由后向前检查，依次把选定系统的资料送入有关单元；

. 令 $CE=4，以便从 C 系统开始由后向前检查当前发声声部是否有效；

. 判断 $C9，若 $C9>H7F 则表示演奏结束，直接返回；在此 $C9 单元起着演奏开关的作用；当前 $C9=7，故执行演奏处理；

. 调 $9AB8 工作子程序，从七页读取弦乐资料进行相关处理；

. 从 C 系统开始，每处理一个系统令 $CE÷2、$CA－1，以指向下一个系统，三个系统全部处理完毕，返回中断控制程序。

. 由于当前只有 $CB=1，故 $CA 递减到零、$CE 递减到 1。

3. 工作子程序的处理

$9AB8 是中断工作子程序的入口，弦乐资料的处理由该段子程序完成。对于示例，工作过程如下：

. 令 $CB－1=0、$91=$C6=0、$C2=0、Y=0，以 $91、$92 为从七页读数地址指针、以 Y 寄存器变址，开始从七页读数；

. 首先、读出音色定义符 Y，有 A=H59，故转 $9B6F，读出音色参数 2 经移位处理变为 H40 送入 $D2 单元（D2、D4 为三个系统的音色数据存储器），返回 $9ACC 继续读数；

. 随后、读出音量、音长定义符 M，有 A=H4D，故转 $9AED，读出 M 后的参数进行检查，若不是 0、1 则返回；若是 0 则表示定义音量，故令 $CF=H30（$CF～$D1 为三系统音量、音长数据存储器）；若是 1 则表示定义音长，故令 $CF=0；

. 读出音量、音长值定义符 V，有 A=H56，故转 $9B3D，读出 V 后的参数进行判断：若是“:”或小于零则返回；若等于 1 再读数检查该参数是否为两位数，若是两位数则把其换算成十六进制数；若第二位数大于 5 则把该参数作为 1 对待；把参数存入 $BC 单元（$BC～$BE 为音量音长参数存储器）；

· 读出音长定义数据，M1V8 的处理过程与 M1V9 相同（从略）；

· 读出节拍定义符 T，有 A＝H54，故转＄9B22，读出 T 后的参数进行检查：若大于 8 或小于 1 说明节拍定义有错，则结束演奏处理；若符合要求则送入＄B9 单元（＄B9～＄BB 为三个系统的节拍存储器）；

· 读出八度音定义符 O，有 A＝H4F，故转＄9B09，读出 O 后的参数进行检查：若大于 5 或小于 0 说明节拍定义有错，则结束演奏处理；若符合要求则送入＄BF 单元（＄BF～＄C1 为三个系统的八度音数据存储器）；

· 读出第一个音符 C，有 A＝H43，当读出的数据为 H41～H47（对应大写字母 A、B、C、D、E、F、G）时，表示当前数据是音符，故均转＄9B83，调＄9C0E 取出音符的时值进行检查：若时值小于零或大于九都说明定义有错，则直接返回；

若时值符合要求则送入 X 寄存器，以 X 变址从音符音长数据区＄8096～＄809F 查出与之对应的音长换算值存入＄C3（＄C3～＄C5 为三系统音符音长换算值存储器），然后令＄28＝＄C3、＄2C＝＄B9，调乘法运算子程序＄8B5C 进行乘法运算，即令音符音长值与节拍值相乘，返回后把乘积送入＄CB（＄CB～＄CD 为三系统的音符发音长度存储器），并令＄C6 暂存＄91 单元的地址指针；

令 V、M、Y 三参数相或后存入＄400 单元；

以 O 参数为偏移量，从＄80A7 查出数据 N1，以音符的 ASCII 码减去 H41 为偏移量，从＄80A0 查出数据 N2，令 N3＝N1＋N2＋＄C2，然后以 N3 为音符的音调数据地址偏移量，从＄8006 音调数据区查出当前音符对应的音调细调值送入第一主声部的＄4002 单元、查出音调粗调值送入＄4003 单元，由于在 PLAY 保留字的处理中已对＄4000、＄4001 两单元赋值，故现在第一主声部开始发出“1”音；

· 返回主控子程序，现在因＄CA－1＝HFF，故返回中断主控程序，恢复 X、Y 寄存器的现场，结束本次中断处理。

以后各音符的处理过程与上相同。其它两个系统的演奏过程留给读者分析。

4·12·3 动作卡通的运动控制

我们知道，F BASIC 在执行 DEF MOVE、POSITION 语句后，仅把定义动作卡通的有关参数写入六页，并不能使卡通运动，只有把有关数据送入二页才能令卡通移动，这一送数操作是在中断处理中完成的。动作卡通的运动控制中断处理程序在系统 ROM 区的＄BE33～＄BFDD，该段程序共分三大段。第一段为中断处理主控程序＄BE33～＄BE43；第二段为动作卡通运动状态检查控制子程序＄BE44～＄BE67；第三段为卡通运动控制工作程序＄BE7B～＄BFDD；其中还有一段动作卡通运动结束处理程序＄BE68～＄BE7A。下面仅以定义 1＃动作卡通为例讨论中断处理工作过程。

例：定义 0＃动作 SP 马里欧哥哥为 1＃卡通，从坐标（64，96）开始以 1/15 秒移动一点的速度向上运动 3 点。

根据题义，对应的定义语句为：

```
10 DEF MOVE(1)=SPRITE(0,1,2,3,0,0)
20 POSITION 1,64,96
30 MOVE 1
```

当仅执行前两行语句时，1＃动作卡通在六页、二页的有关单元数据如下：

646 00

6C8 01 00 02 04 02 04 40 60

290 EF EF EF EF EF EF EF EF EF EF EF EF EF EF EF EF

注——64 的十六进制数为 H40，96 的十六进制数为 H60。

当执行 MOVE 1 后则令＄646＝H10、＄AC＝HFF，这时在执行中断程序时将令＄AB＝HFF，在下一次中断时将转＄BE33，进入动作卡通的中断处理。工作过程如下：

1. 执行＄BE33 保护 X、Y 寄存器的现场；然后调动作卡通运动状态检查控制子程序＄BE44，进行检查判断处理；返回后恢复 X、Y 寄存器的原状态，中断返回。

2. 执行＄BE44，令＄5D＝8 作为计数器，依次检查八个动作卡通的当前运动状态，即取＄645～＄64D 的数据与 H10 进行与运算，若不为零则说明该号卡通已经启动（即由 MOVE 指令作用），于是转＄BE7B 卡通运动工作程序处理；一个卡通处理完毕再检查下一个卡通，直到八个卡通全部检查处理完毕，返回主控程序。

3. 对于本例，当＄A9＝1 时有＄646 AND H10 ≠0，故转＄BE7B 进入动作卡通的运动控制处理。

. 取＄A9 的数据左移三次等于 8，送入＄A6 单元，令 X＝＄A6，令速度计数器＄6CA 的内容减一，若不等于零则转＄BE5A 检查后面的卡通；若等于零则对速度计数器重新赋值，即令＄6CA＝＄6CC＝2；再令运动量计数器＄6CB 减一，检查＄6CB，若等于零则进行运动结束处理；若不为零则进行运动卡通的定义处理。

. 执行＄BE93，以＄6C9 AND H0F 后的数值变址，从＄BC95 数据区取数对＄DD 单元赋值，有＄DD＝0；再从＄BC75 数据区取数置入＄A7、＄A8，使其指向＄BCA5；再以卡通的运动方向数据＄6C8 变址，从＄BD17 数据区取数与卡通运动起点的 X 座标值相加后送入＄6CE，以改变卡通的 X 座标；从＄BD18 数据区取数与卡通运动起点的 Y 座标值相加后送入＄6CF，以改变卡通的 Y 座标；通过以上处理有＄6CE＝H40、＄6CF＝H5E。

. 以 ＄A7、＄A8 为地址指针，取数送入＄D5、＄D6，再以＄D5、＄D6 为地址指针，取数送入＄D7 单元以便决定卡通块的序号。通过以上处理有＄6C9＝H10。

. 以＄A7、＄A8 为地址指针，以 Y＝2 变址，取数送入＄D9、＄DA；再以＄6C8 变址，以＄D9、DA 为地址指针取数对＄DB、＄DC 赋值。以决定当前卡通姿式的四块卡通块的序号组合数据地址。

. 执行＄BF16，检查卡通的运动方向：若不是向上或向下运动则转＄BF71；本例是向上运动，则执行＄BF21 判断卡通种类：若是马里欧、丽莎则转＄BF4B；若是企鹅则执行＄BF30～＄BF4A 的程序段；若是其它卡通则转＄BF71（这是由于在上、下运动时，马里欧、丽莎的运动方式为一类，企鹅的运动方式自成一类，其它的卡通为另一类的缘故。其它方向的运动类型都是相同的。）；因本例的卡通是马里欧，属第一类，故执行＄BF4B 的程序段，调整＄646＝H50，并令＄D7＝0。

. 执行＄BF71，以 Y 变址，以＄DB、＄DC 为基址，取出第一个卡通块的序号加＄DD、＄D7 后送入＄291（第一个卡通块的序号单元）；取出第二个卡通块的序号加＄DD、＄D7 后送入＄295（第二个卡通块的序号单元）；取出第三个卡通块的序号加＄DD、＄D7 后送入＄299（第三个卡通块的序号单元）；取出第四个卡通块的序号加＄DD、＄D7 后送入＄29D（第四个卡通块的序号单元）。

. 执行＄BFA4，以＄A9 变址，从＄646 单元取出卡通状态数据送入＄292、＄296、＄29A、

$ 29E。

· 执行 $ BFB5,以 $ A6 变址,从 $ 6CE 单元取出卡通的 X 座标数据送入 $ 293、$ 29B,令 X 座标值加 8 送入 $ 297、$ 29F。

· 执行 $ BFC9,以 $ A6 变址,从 $ 6CF 单元取出卡通的 Y 座标数据送入 $ 290、$ 294,令 Y 座标值加 8 送入 $ 298、$ 29C。

· 通过以上处理,1 # 卡通的二页定义区 $ 290～ $ 29F 单元的内容如下:

$ 290=5E	$ 294=5E
$ 291=15	$ 295=14
$ 292=50	$ 296=50
$ 293=40	$ 297=48
$ 298=66	$ 29C=66
$ 299=17	$ 29D=16
$ 29A=50	$ 29E=50
$ 29B=40	$ 29F=48

当一个卡通定义完后,返回 $ BE5A 检查定义另外的卡通。

从以上数据可见,执行 MOVE 指令后,该卡通已经向上运动了一点,而且姿式已左右翻转。只要继续跟踪运行,将会证实卡通的运动控制就是不断的调换卡通的定义座标及姿式,这已较好的解释了任天堂游戏中卡通的定义方法与过程,以后的章节还要专题讨论这一问题。

第五章　游戏编程特技

前四章我们概括地讨论了电脑游戏机工作系统的特点，它是我们研究游戏编程特技的理论基础。本章将针对背景、卡通、音响三大游戏组成要素，具体讨论在游戏编程中的实现方法。这些方法也同样适用于各种应用程序的编写。为适应不同程度朋友的需要，本章介绍的程序，开始时均以机器语言、F BASIC 语言两种方式列出程序清单，从第三节开始将只列出机器语言程序清单，读者可对照自己编写写入程序。在机器语言编程中一般使用浮动地址，并使其成为独立的子程序，以方便调用。

5・1　F BASIC 程序与机器语言程序的结合

由于目前流行的电脑游戏机都没有配备监控系统，不能直接编写、运行机器语言程序，故所有机器语言程序的写入、运行只能在 F BASIC 工作状态下进行，这样就出现了两种程序的结合、兼容问题。处理不好轻则程序不能运行，重则造成死机，使辛苦编写、输入的程序毁于一旦，这实在是一件令人沮丧的事情。因此，编程前要首先处理好这一问题。

5・1・1　机器语言程序的写入

在第二章我们曾涉及到机器语言程序的写入问题，本节我们将详细讨论两种机器语言程序写入内存方法的优化。

一、写入内存地址的确定

BS. 2A 版本留给用户的程序存储空间从 $703E～$7FFF 共 4030 个内存单元，$703E 为 F BASIC 暂缓执行型程序的存储首址，这就意味着机器语言程序也只能写入这一空间。通常为避免机器语言程序与 F BASIC 程序的冲突，一般把机器语言程序尽量往后放，具体存放地址可用倒推法算出。如机器语言程序长度为 500 个字节，换算为十六进制数为 H01F4，用 H7FFF 减 H01F4 等于 H7E0C，我们可以选定机器语言程序存放首址为 $7E00。当 F BASIC 程序较长时，可在程序的第一行加一条语句“CLEAR &H××××”，规定 F BASIC 程序区的末址，以避免 F BASIC 程序覆盖了机器语言程序。

总之，确定机器语言程序写入内存地址的原则是：以两种语言程序没有冲突为前提，尽量向后选取。

二、使用 POKE 语句写入内存的方法

以往的 BASIC 语言中的 POKE 语句一次只能向指定地址单元写入一个数据，而 F BASIC 语言中的 POKE 语句则一次可向指定地址写入多个数据，这也是 F BASIC 语言的又一特点。下面通过画面右移子程序 No. 5－1 介绍这一方法的应用。

No. 5－1 画面右移子程序

```
START   PHA
        LDX #$00
        LDY #$00
```

```
MOVE        STX  $2005
            STY  $2005
            LDA  # $FF
YANSHI      STA  $F3       ;以$F3为延时计数器
            JSR  $86B9     ;调STOP键扫描子程序作为短延时
            DEC  $F3
            BNE YANSHI
            INX
            BNE MOVE
            PLA
            RTS
```

如果选定的存放地址为 $7FE0,则 No. 5－1 对应的 6527 机器语言程序如 No. 5－2

No. 5－2

```
7FE0  48        PHA
7FE1  A2 00     LDX  # $00
7FE3  A0 00     LDY  # $00
7FE5  8E 05 20  STX  $2005
7FE8  8C 05 20  STY  $2005
7FEB  A9 FF     LDA  $FF
7FED  85 F3     STA  $F3
7FEF  20 B9 86  JSR  $86B9
7FF2  C6 F3     DEC  $F3
7FF4  D0 F9     BNE  $7FED
7FF6  E8        INX
7FF7  D0 EC     BNE  $7FE5
7FF9  68        PLA
7FFA  60        RTS
```

把程序 No. 5－2 写入内存的 F BASIC 程序为 No. 5－3。

No. 5－3

```
POKE &7FE0,&48,&A2,0,&A0,0,&8E,5,&20,&8C,5,&20,&A9,&FF,&85,&F3, &20,&B9,&86,
&C6,&F3,&D0,&F9,&E8,&D0,&EC,&68,&60
```

No. 5－3 中所有十六进制数中的“&H”均省略了“H”，这是 F BASIC 允许的一种简写形式。

由以上讨论可见，一行 POKE 语句最多可一次写入 60 个数据，当编写暂缓执行型程序时则最多可写 48 个数据(由于程序列表时每行 256 个字符的限制)。

三、使用循环语句写入内存的方法

用循环语句把程序 No. 5－2 写入内存的 F BASIC 程序见 No. 5－4。

```
5 REM No. 5－4 画面右移子程序
10 FOR I=0 TO 26
20 READ A ; POKE &7FE0 + I, A
30 NEXT
40 DATA &48, &A2, 0, &A0, 0, &8E, 5, &20, &8C, 5,&20, &A9, &FF, &85, &F3, &20, &B9, &86,
   &C6,&F3, &D0, &F9, &E8, &D0, &EC, &68, &60
```

通过使用 No. 5－3、No. 5－4 把机器语言程序写入内存的实践，都会产生一种别扭的感觉，原因是：每个数据前面都要带一个前缀"&"，它一方面影响了程序的输入速度；另一方面使机器语言程序变得不直观。若象程序 No. 2－6 那样把十六进制数据改为十进制数，虽然可以提高程序的输入速度，但是将使机器语言程序变得更加难读，不利程序的修改。那么是否有两全其美的方法呢？程序 No. 5－5 较好的解决了这一问题。

```
5 REM No. 5－5 画面右移子程序
10 FOR I＝0 TO 26
20 READ A$ ：POKE &7FE0 ＋ I, VAL("&" ＋ A$)
30 NEXT
40 DATA 48, A2, 0, A0, 0, 8E, 5, 20, 8C, 5, 20,A9, FF, 85, F3, 20, B9, 86, C6, F3, D0, F9,E8, D0,
   EC, 68, 60
```

程序 No. 5－5 中，先把数据当作字符从数据区读出，然后再把字符转化为数值写入内存，从而解决了数据输入方便与数据读、改直观的矛盾。因此，程序 No. 5－5 是最优化的机器语言程序写入方法。也是最常用的机器语言程序写入方法。一般来说，当程序量较小时使用 No. 6－3；当程序量较大时多使用 No. 5－5。

5·1·2　机器语言程序的运行

F BASIC 的保留字中有一条未公布的指令"CALL"，它的功能就是调用机器语言子程序，其语句格式为

CALL ××××（机器语言子程序首地址）

现在我们可以进行画面右移操作了。在程序 No. 5－3（或 No. 5－4、No. 5－5）运行一次后，即可把程序删去。只要键入"CALL &7FE0"即可令画面连续右移。改变 $F3 的初值可调整画面的右移速度；改变 X 寄存器的初值可以控制画面右移的距离；若改"INX"为"DEX"，则可令画面左移；若改"INX"为"INY"或"DEY"，则又可令画面上、下移动。因此，程序 No. 5－1 是一个画面位移通用子程序。

5·1·3　应用举例——真正的电子钟

在中华学习机、PC 机中都有一个时钟程序，用户可以即时校对当前时间，可以随意控制时钟的显示与关闭，使用非常方便。电脑游戏机没有配备时钟软件，以往虽然有介绍模拟电子钟的程序，但由于是用 F BASIC 语言编写的，因而与其它 F BASIC 程序相冲突，不能并行工作，电子钟工作时则不能运行其它程序。另外电子钟的时间取样是通过延时语句获得的，故走时极不精确。所以，这类电子钟的实用价值不大。

在掌握了机器语言程序的写入和运行方法后，我们可着手编写真正的电子钟程序了。编程思路与编程过程如下：

1、以中断周期为时间基准，由于一个中断周期为 0.02 秒，那么只要设置一个中断计数器，就可以得到精确的时基；

2、用机器语言编程，程序放在中断中处理，这样可以避免与用户程序的冲突，节省用户的程序存储空间。

3、为了避免过多的现场保护，我们选取 $7FF0～$7FF3 作为计数、计秒、计分、计时器，以避开零页单元的使用。

4、选定＄F3 为电子钟的显示开关，＄F3=0 关显示；＄F3=1 开显示。

5、为避免电子钟的时间显示破坏用户画面，故把电子钟的时间显示座标选定在 F BASIC 显示画面之外。

6、机器语言程序清单

根据以上编程思路，编写机器语言程序如 No. 5－6(选定程序存放首址为＄7E5F)

No. 5－6 电子钟

```
7E5F  A5 9D     LDA $9D       ;画面定位子程序
7E61  8D 05 20  STA $2005
7E64  A5 69     LDA $69
7E66  8D 05 20  STA $2005
7E69  60        RTS
7E6A  A5 33     LDA $33       ;开显示子程序
7E6C  8D 01 20  STA $2001
7E6F  60        RTS
7E70  8D 06 20  STA $2006     ;置 PPU 地址子程序
7E73  8E 06 20  STX $2006
7E76  60        RTS
7E77  20 80 7E  JSR $7E80     ;关显示子程序
7E7A  A9 00     LDA $00
7E7C  8D 01 20  STA $2001
7D7F  60        RTS
7E80  AD 02 20  LDA $2002
7E83  30 FB     BMI $7E80
7E85  AD 02 20  LDA $2002
7E88  10 FB     BPL $7E85
7E8A  60        RTS
7E8B  18        CLC           ;加法子程序
7E8C  A5 2C     LDA $2C
7E8E  65 28     ADC $28
7E90  85 28     STA $28
7E92  A5 2D     LDA $2D
7E94  65 29     ADC $29
7E96  85 29     STA $29
7E98  60        RTS
7E99  38        SEC           ;减法子程序
7E9A  A5 28     LDA $28
7E9C  E5 2C     SBC $2C
7E9E  85 28     STA $28
7EA0  A5 29     LDA $29
7EA2  E5 2D     SBC $2D
7EA4  85 29     STA $29
7EA6  60        RTS
7EA7  A9 00     LDA #$00
```

```
7EA9  85 46      STA $46
7EAB  A5 46      LDA $46
7EAD  48         PHA
7EAE  A9 10      LDA #$10     ;转换万位送入$3D单元
7EB0  85 2C      STA $2C
7EB2  A9 27      LDA #$27
7EB4  85 2D      STA $2D
7EB6  20 F3 7E   JSR $7EF3
7EB9  A9 E8      LDA #$E8     ;转换千位送入$3E单元
7EBB  85 2C      STA $2C
7EBD  A9 03      LDA #$03
7EBF  85 2D      STA $2D
7EC1  20 F3 7E   JSR $7EF3
7EC4  A9 64      LDA #$64     ;转换百位送入$3F单元
7EC6  85 2C      STA $2C
7EC8  A9 00      LDA #$00
7ECA  85 2D      STA $2D
7ECC  20 F3 7E   JSR $7EF3
7ECF  A9 0A      LDA #$0A     ;转换十位送入$40单元
7ED1  85 2C      STA $2C
7ED3  20 F3 7E   JSR $7EF3
7ED6  A5 28      LDA $28      ;转换个位送入$41单元
7ED8  18         CLC
7ED9  69 30      ADC #$30
7EDB  20 EC 7E   JSR $7EEC
7EDE  A9 00      LDA #$00
7EE0  20 EC 7E   JSR $7EEC
7EE3  68         PLA
7EE4  85 46      STA $46
7EE6  60         RTS
7EE7  20 8B 7E   JSR $7E8B    ;把转换后的十进制数送入$3D～$41
7EEA  A5 47      LDA $47
7EEC  A6 46      LDX $46
7EEE  95 3D      STA $3D,X
7EF0  E6 46      INC $46
7FF2  60         RTS
7FF3  A9 30      LDA #$30
7FF5  85 47      STA $47
7FF7  20 99 7E   JSR $7E99
7FFA  90 EB      BCC $7EE7
7FFC  E6 47      INC $47
7FFE  4C F7 7E   JMP $7EF7
```

电子钟主控程序

```
7F01  A9 0A      LDA #$0A     ;设置中断入口
```

```
7F03  85 EE      STA  $EE
7F05  A9 7F      LDA  # $7F
7F07  85 EF      STA  $EF
7F09  60         RTS
7F0A  48         PHA
7F0B  20 12 7F   JSR  $7F12
7F0E  68         PLA
7F0F  4C 0F 88   JMP  $880F
7F12  EE F0 7F   INC  $7FF0      ;计数
7F14  AD F0 7F   LDA  $7FF0
7F16  C9 32      CMP  # $32
7F18  F0 01      BEQ  $7F1D
7F1A  60         RTS
7F1D  A9 00      LDA  # $00
7F1F  8D F0 7F   STA  $7FF0
7F22  EE F1 7F   INC  $7FF1      ;计秒
7F25  AD F1 7F   LDA  $7FF1
7F28  C9 3C      CMP  # $3C
7F2A  D0 23      BNE  $7F4F
7F2C  A9 00      LDA  # $00
7F2E  8D F1 7F   STA  $7FF1
7F31  EE F2 7F   INC  $7FF2      ;计分
7F34  AD F2 7F   LDA  $7FF2
7F37  C9 3C      CMP  # $3C
7F39  D0 14      BNE  $7F4F
7F3B  A9 00      LDA  # $00
7F3D  8D F2 7F   STA  $7FF2
7F40  EE F3 7F   INC  $7FF3      ;计时
7F43  AD F3 7F   LDA  $7FF3
7F46  C9 0D      CMP  # $0D
7F48  90 05      BCC  $7F4F
7F4A  A9 00      LDA  # $00
7F4C  8D F3 7F   STA  $7FF3
7F4F  98         TYA             ;保护现场
7F50  48         PHA
7F51  8A         TXA
7F52  48         PHA
7F53  A2 1F      LDX  # $1F
7F55  B5 28      LDA  $28,X
7F57  48         PHA
7F58  CA         DEX
7F59  10 FA      BPL  $7F55
7F5B  A0 00      LDY  # $00      ;把十进制数转换为可显示的 ASCII 码
7F5D  A9 02      LDA  # $02      ;存入 $3D～$41
```

```
7F5F  8D F4 7F   STA $7FF4
7F62  AE F4 7F   LDX $7FF4
7F65  BD F1 7F   LDA $7FF1,X
7F68  85 28      STA $28
7F6A  A9 00      LDA #$00
7F6C  85 29      STA $29
7F6E  20 A7 7E   JSR $7EA7
7F71  A5 40      LDA $40        ;把时间数据存入$7FF5～$7FFD
7F73  99 F5 7F   STA $7FF5,Y
7F76  C8         INY
7F77  A5 41      LDA $41
7F79  99 F5 7F   STA $7FF5,Y
7F7C  C8         INY
7F7D  A9 3A      LDA #$3A
7F7F  99 F5 7F   STA $7FF5,Y
7F82  C8         INY
7F83  CE F4 7F   DEC $7FF4
7F86  AE F4 7F   LDX $7FF4
7F89  10 DA      BPL $7F65
7F8B  A5 93      LDA $93
7F8D  F0 1D      BEQ $7FAC
7F8F  20 77 7E   JSR $7E77
7F92  A0 00      LDY #$00
7F94  A9 20      LDA #$20
7F96  A2 36      LDX #$36
7F98  20 70 7E   JSR $7E70
7F9B  B9 F5 7F   LDA $7FF5,Y    ;送显时间
7F9E  8D 07 20   STA $2007
7FA1  C8         INY
7FA2  C0 08      CPY #$08
7FA4  D0 F5      BNE $7F9B
7FA6  20 6A 7E   JSR $7E6A
7FA9  20 5F 7E   JSR $7E5F
7FAC  A2 00      LDX #$00       ;恢复现场
7FAE  68         PLA
7FAF  95 28      STA $28,X
7FB1  E8         INX
7FB2  E0 20      CPX #$20
7FB4  D0 F8      BNE $7FAE
7FB6  68         PLA
7FB7  AA         TAX
7FB8  68         PLA
7FB9  A8         TAY
7FBA  60         RTS
```

7、电子钟的写入程序

程序 No. 5－6 的 F BASIC 写入程序如 No. 5－7

```
5 REM No. 5—7 电子钟写入程序
10 FOR I=0 TO 347 : READ A$
20 POKE &7E5F + I, VAL( "&" + A$)
30 NEXT
40 DATA A5, 9D, 8D, 05, 20, A5, 69, 8D, 05, 20,60, A5, 33, 8D, 01, 20, 60, 8D, 06, 20, 8E,06, 20,
   60, 20, 80, 7E, A9, 00, 8D, 01, 20,60, AD, 02, 20, 30, FB, AD, 02, 20, 10, FB,60, 18, A5, 2C,
   65, 28, 85, 28, A5, 2D, 65,29, 85, 29, 60
50 DATA 38, A5, 28, E5, 2C, 85, 28, A5, 29, E5,2D, 85, 29, 60, A9, 00, 85, 46, A5, 46, 48,A9, 10,
   85, 2C, A9, 27, 85, 2D, 20, F3, 7E,A9, E8, 85, 2C, A9, 03, 85, 2D, 20, F3, 7E,A9, 64, 85, 2C,
   A9, 00, 85, 2D, 20, F3, 7E,A9, 0A, 85, 2C, 20, F3, 7E
60 DATA A5, 28, 18, 69, 30, 20, EC, 7E, A9, 00,20, EC, 7E, 68, 85, 46, 60, 20, 8B, 7E, A5,47, A6,
   46, 95, 3D, E6, 46, 60, A9, 30, 85,47, 20, 99, 7E, 90, EB, E6, 47, 4C, F7, 7E
70 DATA A9, 0A, 85, EE, A9, 7F, 85, EF, 60, 48,20, 12, 7F, 68, 4C, 0F, 88, EE, F0, 7F, AD,F0,
   7F, C9, 32, F0, 01, 60, A9, 00, 8D, F0,7F, EE, F1, 7F, AD, F1, 7F, C9, 3C, D0, 23,A9, 00, 8D,
   F1, 7F, EE, F2, 7F, AD, F2, 7F,C9, 3C, D0, 14, A9, 00, 8D, F2, 7F
80 DATA EE, F3, 7F, AD, F3, 7F, C9, 0D, 90, 05,A9, 00, 8D, F3, 7F, 98, 48, 8A, 48, A2, 1F,B5,
   28, 48, CA, 10, FA, A0, 00, A9, 02, 8D,F4, 7F, AE, F4, 7F, BD, F1, 7F, 85, 28, A9,00, 85, 29,
   20, A7, 7E, A5, 40, 99, F5, 7F,C8, A5, 41, 99, F5, 7F, C8, A9, 3A, 99, F5,7F, C8, CE, F4, 7F,
   AE, F4, 7F, 10, DA
90 DATA A5, 93, F0, 1D, 20, 77, 7E, A0, 00, A9,20, A2, 36, 20, 70, 7E, B9, F5, 7F, 8D, 07,20,
   C8, C0, 08, D0, F5, 20, 6A, 7E, 20, 5F,7E, A2, 00, 68, 95, 28, E8, E0, 20, D0, F8,68, AA, 68,
   A8, 60
```

8、电子钟的运行

程序 No. 5－7 输入完毕,检查无误后,运行一次可以删除，以输入其它程序。也可以暂不删除,以便有错时修改。

电子钟的 F BASIC 运行程序见 No. 5—8

```
5 REM No. 5—8 电子钟运行程序
100 CALL 32513 : POKE 147, 1 :CLS                  (启动电子钟)
110 INPUT "INPUT (hour) "; A : POKE 32755,A        ( 调时 )
120 INPUT "INPUT (minute)";A : POKE 32754,A        ( 调分 )
130 INPUT "INPUT (second)";A : POKE 32753,A        ( 调秒 )
140 CLS : LOCATE 5,5 : PRINT "WELCOME USING";: LOCATE 4 , 7 :PRINT "ELECTRONIC
    CLOCK"
150 PAUSE 300: CLS : NEW
```

运行程序 No. 5—8 后,根据屏幕提示依次输入当前的时间“时、分、秒”,电子钟即进入计时运行。运行中欲停止电子钟的显示可执行“POKE 147,0”;若开显示可执行“POKE 147,1”;当欲改变计时的初始时间时可执行“POKE 32755,××”以改变计时、“POKE 32754,××”以改变计分、“POLE 32753,××”以改变计秒。从而可以使用它检测程序的运行时间等,使用十分方便。

9、电子钟的特点

• 走时精度高,可与电子石英钟相媲美。

·与 F BASIC 程序并行工作，无论进行 F BASIC 编程或运行程序，都不影响电子钟的走时精度和显示。

·可随时进行时间校对，故即可作电子钟使用又可用于电脑工作计时，还可用于程序运行时间的检测。

·电子钟的时间显示在屏幕的右上角，在 F BASIC 显示画面之外，故不破坏 F BASIC 的显示画面。

10、注意事项

·有些 F BASIC 系统软件的工作中使用了＄7F00～＄7FFF 用户 RAM 空间（如裕兴键盘等），这时已不能照搬程序 No. 5－7、No. 5－8，必须避开这一空间。如把电子钟程序安装在＄7D00 开始的 RAM 段或其它空间。相应地＄7FF0～＄7FF3 计时单元也要调整。

·对于使用 RAM 替作背景、卡通字模（图形库）存储器 ROM 的工作系统，在电子钟运行时，有可能干扰背景图形的空格字模数据，造成画面显示紊乱。这时可盲打运行以下命令，即可恢复正常。

```
POKE &2006,2 : POKE &2006,0 :
POKE &2007,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

·程序 No. 5－7、No. 5－8 虽然是针对 BS. 2A 版本的 F BASIC 工作系统编写的，但只要更改中断返回入口可适用于任何 F BASIC 版本。

5·2　系统子程序简介

BS. 2A 版本 F BASIC 工作系统的解释程序中，有许多系统子程序可供我们在编程中调用。通过调用这些系统子程序，可以极大的简化我们编写的应用程序，这一点将在本节最后列出的电子钟简化程序中充分体现。

5·2·1　移位类子程序

1、＄EB3F

功能：令＄8E～＄9F 单元的内容循环前移一个单元。

注意：保护＄00、＄01 单元，X、Y 寄存器的现场。

2、＄EB34

功能：令＄8E～＄9F 单元的内容循环后移一个单元。

注意：保护＄00、＄01、X、Y。

3、＄EB76

功能：令连续的八个单元的内容循环前移一个单元。

注意：调用前要对＄00～＄01 置入移位单元的首址；保护 X、Y。

4、＄EB87

功能：令连续的 16 个单元的内容前移一个单元。

注意：调用前要对＄00～＄01 置入移位单元的首址；保护 X、Y。

5、＄EB98

功能：令连续的八个单元的内容后移一个单元。

注意：调用前要对＄00～＄01 置入移位单元的首址；保护 X、Y。

6、$EBAF

功能：令连续的16个单元的内容后移一个单元。

注意：调用前要对$00～$01置入移位单元的首址；保护X、Y。

5·2·2 数制转换类子程序

1、$8CF9

功能：把用ASCII码表示的十进制数转换成十六进制数值存于$28、$29单元中。数据地址由$09、$0A指出。

注意：数据的ASCII码要预先存入RAM区，并以“0”为结束标志，令$09、$0A指向数据首址。

2、$8D3C

功能：把存于$28、$29单元的十六进制数值转换成十进制数的ASCII码送入$3D～$41中，以便输出。

注意：把数据预先置入$28、$29单元。

3、$8DD8

功能：把$28、$29中的16进制数值转换为16进制数的ASCII码存入$3D～$40单元，令$41=0。

4、$8DEF

功能：把ASCII码形式的16进制数还原为16进制数值存入$28、$29单元中。

注意：要求$09、$0A指向数据存储区首址，并以“0”为数据区结束标志。

5、$8CE9

功能：对存入$3D～$41单元中待输出的数据整形，删除数据前的0。

5·2·3 置数、送数类子程序

1、$8B47

功能：令$35～$3C=0

2、$A1CD

功能：令$29=0、$2D=0。

3、$A2BA

功能：令$28=$29=$2A=0。

4、$EC4C

功能：令$8E～$9F=0。

5、$DF8F

功能：令$7600～$76FF=0。

注意：保护A、Y。

6、$A6DF

功能：令$200～$2FF=HEF。

注意：保护A、X。

7、$98A1

功能：令$580～$58F=H20、$590=0。

注意：保护 X、Y、A。

8、$ 8A8D

功能：把 $ 2C、$ 2D、$ 2E 单元中的内容送入 $ 28、$ 29、$ 2A 单元中。

9、$ 8FF8

功能：保存 $ 09、$ 0A 单元的地址指针于 $ 9A、$ 9B 单元中。$ 09→$ 9A、$ 0A→$ 9B。

10、$ 9001

功能：送还 $ 09、$ 0A 单元的地址指针。$ 9A→$ 09、$ 9B→$ 0A。

11、$ A0B0

功能：$ 28→$ 8C 、$ 29→$ 8D

12、$ A007

功能：$ 8E→$ 8C、$ 8F→$ 8D

13、$ 98B4

功能：令 $ 580～$ 58F → $ 501～$ 510，并令 $ 511＝0。

注意：保护 X、Y、A。

14、$ DF9A

功能：把 $ 63D～$ 714 单元的内容送入 $ 7100～$ 71D7。

注意：保护 A、X。

5·2·4 运算类子程序

1、$ 8B0D

功能：令 $ 28、$ 29 的内容与 $ 2C、$ 2D 相减，差存于 $ 28、$ 29 中。

过程：C＝1、$ 28－$ 2C→$ 28、$ 29－$ 2D→$ 29

2、$ 8B39

功能：令 $ 28、$ 29 的内容与 $ 2C、$ 2D 相加，和存于 $ 28、$ 29。

过程：C＝0、$ 28＋$ 2C→$ 28、$ 29＋$ 2D→$ 29。

3、$ 8BB5

功能：令 $ 28、$ 29÷$ 2C、$ 2D，商存于 $ 28、$ 29。

注意：保护 $ 28～$ 2D、$ 34、A、X、P。

4、$ 8B5C

功能：令 $ 28、$ 29×$ 2C、$ 2D，积存于 $ 28、$ 29。

注意：保护 $ 28～$ 2D、$ 34～$ 3C、A、X、P。

5、$ 8A7C

功能：对 $ 28、$ 29 单元的内容求补码，结果仍存 $ 28、$ 29 中。

过程：C＝1、0－$ 28→$ 28、0－$ 29→$ 29。

6、$ 8A61

功能：对 $ 2C、$ 2D 单元中的内容求补码，结果仍存于 $ 2C、$ 2D 中。

过程：C＝1、0－$ 2C→$ 2C、0－$ 2D→$ 2D。

7、$ 8BF7

功能：比较 $ 29、$ 28 与 $ 2D、$ 2C 的大小。

8、$ 8C09

功能:比较＄28、＄29与＄2C、＄2D的大小:

若＄28、＄29 = ＄2C、＄2D则令C=1、A=0,返回;

若＄28、＄29 > ＄2C、＄2D则令C=1、A=HFF,返回;

若＄28、＄29 < ＄2C、＄2D则令C=0、A=HFF,返回。

9、＄8CDD

功能:比较A寄存器中的数据:

若等于H30~H39则令C=0,返回;

否则令C=1返回。

10、＄98CC

功能:比较＄501~＄511与＄580~＄590的内容,若相同则令C=0返回;若不同则令C=1返回。

注意:保护A、X、Y。

11、＄8E22

功能:令＄28、＄29中的数据扩大16倍。

注意:保护乘法运算的现场。

12、＄A341

功能:令A寄存器中的内容乘8后,再送入X。

13、＄A6C5

功能:令A寄存器中的内容连续左移位四次。

14、＄EDE8

功能:随机数发生器,随机数存于＄51~＄58单元中。

注意:保护＄00单元。调用前要先对＄51单元置入数值。

5·2·5 变址取数类子程序

1、＄8CC7

功能:读数A=＄0A＄09+Y

说明:＄0A、＄09为地址指针,＄0A存地址高位、＄09存地址低位。

2、＄8C9D

功能:读数A=＄0A＄09+Y,＄09+1,若＄09=0则＄0A+1。

3、＄8CD1

功能:先令＄09+1,再读数。A=＄0A＄09+Y,若A=H20则令＄09+1后再读。

4、＄8375

功能:先令＄09+1,再读数。A=＄0A＄09+Y,并令A与H3A比较。

5、＄8E35

功能:读数A=＄4E＄4D+Y 。

说明:＄4E、＄4D为地址指针,＄4E存地址高位、＄4D存地址低位。

6、＄8E32

功能:先令＄4D+1,再读数。A=＄4E＄4D+Y。

7、＄8FA2

功能:读数A=＄55＄54+Y。

说明：$55、$54 为地址指针，$55 存地址高位、$54 存地址低位。

5·2·6 地址指针设置类子程序

1、$8522

功能：令 $19、$1A 地址指针减一，若 $19=0 则令 $19=HFF、$1A−1 。

2、$8516

功能：令 $0B、$0C 地址指针减一，若 $0B=0 则令 $0B=HFF、$0C−1。

3、$8CA1

功能：令 $09、$0A 地址指针加一，若 $09=0 则令 $0A+1。

4、$9927

功能：令 $09、$0A 地址指针加三，若有进位则令 $0A+1。

5、$8FA7

功能：令 $54、$55 地址指针加一，若 $54=0 则令 $55+1。

6、$B3B2

功能：令 $91、$92 地址指针加一，若 $91=0 则令 $92+1。

7、$B3AB

功能：令 $AD、$AE 地址指针加一，若 $AD=0 则令 $AE+1。

5·2·7 字符、图形显示类子程序

1、$87F8

功能：以 $52、$53 为显示内容的数据地址指针，以 Y 变址，先把数据读出送入七页，然后再令 A=1→$63，以显示字符、图形。

条件：· 向 $52、$53 单元置入送显数据区首址，低位在前；

· 向 $62 单元置入使用七页的首址；

· 向 $48、$49 单元置入显示内容的屏幕初始座标。

2、$B254

功能：令光标显示。

注意：作好乘法现场保护。

3、$B25B

功能：令光标隐藏。

注意：同 2。

4、$AE48

功能：在光标处挪出一个空格，相当于按 Ins 键。

注意：作好乘法现场保护，还使用 $6B、$6C、$B0 单元和 A、X、Y 寄存器。

5、$ADB2

功能：删去光标前的一个空格，相当于按 Del 键。

注意：同 4，还使用 $19、$1A 单元。

6、$ADA1

功能：改变光标形状。

要求：调用前把光标形状的代码存入 $E0、$E1 单元。

7、$AD8B

功能:置光标于屏幕左上角。

8、$AD76

功能:令光标右移一格,相当于按→键。

9、$AD64

功能:令光标左移一格,相当于按←键。

10、$AD5A

功能:令光标下移一行,相当于按↓键。

11、$AD53

功能:令光标上移一行,相当于按↑键。

12、$ACF4

功能:清除光标后的所有显示内容,相当于按 Ctrl+Z 键。

注意:保护乘法现场。

13、$ACA1

功能:保存光标的当前座标值于$6B、$6C。

过程:$48→$6B、$49→$6C

14、$ACAA

功能:从$6B、$6C 取回光标的原座标值。

过程:$6B→$48、$6C→$49

15、$AC45

功能:令光标在原位下移一行,若处于最底行则屏幕上滚一行。

注意:保护乘法现场。

16、$AC07

功能:令光标下移一行,位于行首。

17、$ABA6

功能:根据光标的当前座标$48、$49 的数值,计算其对应的 PPU 地存入$4A、$4B。

注意:保护乘法现场。

18、$87F8

功能:把以$52、$53 为地址指针指向的 RAM 区数据取出送入七页显示。

19、$87EE

功能:当向七页送数结束后,再送一个零作为结束标志。

过程:X=$62、A=0、A → $700+X、$62+1

20、$87F0

功能:把 A 寄存器中的内容送入七页。

21、$87B7

功能:以 Y=0 变址,取出$52、$53 指向 RAM 区的内容送入七页,七页以$62 变址。

22、$8761

功能:送显字符串。

要求:· 字符串长度存入$400

· 字符串首址存入$401、$402,低位在前

23、$ 865F

功能:令 $ 52、$ 53 指向 $ 3D～ $ 41 中不为零的单元,然后把 $ 3D～ $ 41 中的内容送入七页显示。

24、$ ABFA

功能:中断开始后开显示

25、$ ABF1

功能:中断开始后关显示

26、$ ABFD

功能:开显示

27、$ ABF4

功能:关显示

28、$ A7C0

功能:开卡通页显示

29、$ A7CC

功能:关卡通页显示

30、$ A9FF

功能:送显"OK"。

31、$ AC34

功能:送显 A 寄存器中的内容

32、$ AC07

功能:令光标下移一行,位于行首。

5・2・8 与 PPU 通信类子程序

1、$ AAFE

功能:写 PPU 前的处理

2、$ ABE6

功能:写 PPU 后的处理

3、$ ABD8

功能:设定 PPU 通信地址

要求:把 PPU 地址高位置入 $ 4B、低位置入 $ 4A。

4、$ 81F8

功能:设定 PPU 通信地址

要求:把 PPU 地址高位置入 A、低位置入 X

5、$ 8876

功能:读出由 $ 4A、$ 4B 指出的 PPU 地址内容的 ASCII 码存入 $ 73 单元。

6、$ 819C

功能:在屏幕上画一个大方框。

7、$ B3F1

功能:开中断

8、$ B3EB

功能:关中断

9、$ EE43

功能:查表法绘图(向 PPU 送数)

要求:数据区首址存于 $ 00、$ 01 中，低位在前。

数据区中第一、第二数据为 PPU 首址,高位在前。

5·2·9 清屏延时类子程序

1、$ AB75

功能:清背景零页

2、$ AB71

功能:清背景一页

3、$ 91B3

功能:延时

要求:调用前要对 $ 400、$ 401 置入延时参数。

$ 400 置 1,相当于 PAUSE 1; $ 401 置 1,相当于 PAUSE 256。

4、$ 86B9

功能:扫描 STOP 键,可作为短延时子程序调用。

5·2·10 发声类子程序

1、$ B314

功能:发"嗒"声,(按键声)。

2、$ B30F

功能:蜂鸣声(BEEP)

3、$ FD5B

功能:用第一主声部发出四种效果音。

说明:对 $ FF 置入 2、4、8、H40

4、$ FE14

功能:用第二主声部演奏背景音乐。

说明:对 $ FC 置入 1、2、3

5、$ FE93

功能:用第三主声部发出音乐片断。

说明:对 $ FE 置入数据。

5·2·11 中断返回类子程序

1、$ 8862

功能:开背景页显示

2、$ 885F

功能:关背景页显示

3、$ 8990

功能:把存入七页的背景配色数据送入 $ 3F××

过程：H3F→＄2006、＄700×4→＄2006、＄701→＄2007、……＄704→＄2007；＄9D、＄69→＄2005；＄32 OR H80→＄2000；＄63=0；＄9D、＄69→＄2005

4、＄89BE

功能：把存入七页的卡通配色数据送入＄3F××

过程：同上

5、＄89CC

功能：执行 Ins 操作。

过程：＄4B、＄4A→＄2006；A=＄2007→＄500+X；＄4B、＄4A→＄2006；A=＄500+X→＄2007；＄9D、＄69→＄2005；0→＄63

6、＄8A02

功能：从＄4B、＄4A 指向的 PPU 地址开始，把连续的 H1C 个显示单元的数据读出存入＄500+X，最后令＄63=H0F。

7、＄8A1A

功能：令＄49+1，从＄500+X 中取数送入＄4B、＄4A 指向的 PPU 显示单元中，连送 H1C 个

8、＄8961

功能：取出＄4B、＄4A 指向 PPU 单元的内容送入五页(共取光标开始到行尾的内容)；把五页的内容送入由＄1A、＄19 指向的 PPU 单元(数量与取出时相同)。

过程：＄4B、＄4A→＄2006；A=＄2007→＄500+X；＄1A、＄19→＄2006；A=＄500+X→＄2007。取数、送数的起始位置由＄48 决定。

9、＄8925

功能：取出与＄49 单元对应的一行内容存入＄500 以后的单元中。

10、＄88AA

功能：取出由＄64 指定的七页内容送入当前光标对应的 PPU 单元中。

说明：当达到右边界时，令＄48=0、＄49+1；

如 ＄49<H18 则中断返回；

如 ＄49=H18 则令显示上卷一行。

11、＄8898

功能：清除从光标开始到行尾的全部内容。

12、＄8890

功能：把＄73 单元的内容送入由＄4B、＄4A 指向的 PPU 中。

13、＄8882

功能：读出由＄4B、＄4A 指定 PPU 的内容存入＄73 单元；把光标形状(＄61)送入 PPU。

过程：＄4B、＄4A→＄2006；A=＄2007，A=＄2007→＄73；＄4B、＄4A→＄2006；＄61→＄2007

5·2·12 读键盘、操纵器类

1、＄B343

功能：读出当前键盘输入的内容，直到按回车键为止。输入的内容存于＄500 开始的内存单元中。

2、＄A682

功能:读 1#操纵器按键,键码存于 $ 90 单元。

3、$ A685

功能:读 0#操纵器按键,键码存于 $ 90 单元。

5·2·13 电子钟程序的优化

在上一节讨论的 No.5—6 的电子钟程序共占用了 348 个字节,如果我们调用背景开、关显示子程序,置屏幕、置 PPU 地址子程序,十六进制数据转换为十进制数据子程序,可使程序 No.5—6 得到极大的简化。简化的电子钟程序见 No.5—9。

No.5—9 电子钟实用程序

```
7F01  A9 0A      LDA # $0A      ;设置中断入口
7F03  85 EE      STA $EE
7F05  A9 7F      LDA # $7F
7F07  85 EF      STA $EF
7F09  60         RTS
7F0A  48         PHA
7F0B  20 12 7F   JSR $7F12
7F0E  68         PLA
7F0F  4C 0F 88   JMP $880F
7F12  EE F0 7F   INC $7FF0      ;计数
7F14  AD F0 7F   LDA $7FF0
7F16  C9 32      CMP # $32
7F18  F0 01      BEQ $7F1D
7F1A  60         RTS
7F1D  A9 00      LDA # $00
7F1F  8D F0 7F   STA $7FF0
7F22  EE F1 7F   INC $7FF1      ;计秒
7F25  AD F1 7F   LDA $7FF1
7F28  C9 3C      CMP # $3C
7F2A  D0 23      BNE $7F4F
7F2C  A9 00      LDA # $00
7F2E  8D F1 7F   STA $7FF1
7F31  EE F2 7F   INC $7FF2      ;计分
7F34  AD F2 7F   LDA $7FF2
7F37  C9 3C      CMP # $3C
7F39  D0 14      BNE $7F4F
7F3B  A9 00      LDA # $00
7F3D  8D F2 7F   STA $7FF2
7F40  EE F3 7F   INC $7FF3      ;计时
7F43  AD F3 7F   LDA $7FF3
7F46  C9 0D      CMP # $0D
7F48  90 05      BCC $7F4F
7F4A  A9 00      LDA # $00
7F4C  8D F3 7F   STA $7FF3
```

```
7F4F  98        TYA              ;保护现场
7F50  48        PHA
7F51  8A        TXA
7F52  48        PHA
7F53  A2 1F     LDX #$1F
7F55  B5 28     LDA $28,X
7F57  48        PHA
7F58  CA        DEX
7F59  10 FA     BPL $7F55
7F5B  A0 00     LDY #$00         ;把十进制数转换为可显示的 ASCII 码
7F5D  A9 02     LDA #$02         存入 $3D～$41
7F5F  8D F4 7F  STA $7FF4
7F62  AE F4 7F  LDX $7FF4
7F65  BD F1 7F  LDA $7FF1,X
7F68  85 28     STA $28
7F6A  A9 00     LDA #$00
7F6C  85 29     STA $29
7F6E  20 3C 8D  JSR $8D3C
7F71  A5 40     LDA $40          ;把时间数据存入 $7FF5～$7FFD
7F73  99 F5 7F  STA $7FF5,Y
7F76  C8        INY
7F77  A5 41     LDA $41
7F79  99 F5 7F  STA $7FF5,Y
7F7C  C8        INY
7F7D  A9 3A     LDA #$3A
7F7F  99 F5 7F  STA $7FF5,Y
7F82  C8        INY
7F83  CE F4 7F  DEC $7FF4
7F86  AE F4 7F  LDX $7FF4
7F89  10 DA     BPL $7F65
7F8B  A5 93     LDA $93
7F8D  F0 1D     BEQ $7FAC
7F8F  20 F1 AB  JSR $ABF1
7F92  A0 00     LDY #$00
7F94  A9 20     LDA #$20
7F96  A2 36     LDX #$36         ;置 PPU 地址 $2036
7F98  20 F8 81  JSR $81F8
7F9B  B9 F5 7F  LDA $7FF5,Y      ;送显时间
7F9E  8D 07 20  STA $2007
7FA1  C8        INY
7FA2  C0 08     CPY #$08
7FA4  D0 F5     BNE $7F9B
7FA6  20 FD AB  JSR $ABFD
7FA9  20 E6 AB  JSR $ABE6
```

```
7FAC A2 00     LDX #$00        ;恢复现场
7FAE 68        PLA
7FAF 95 28     STA $28,X
7FB1 E8        INX
7FB2 E0 20     CPX #$20
7FB4 D0 F8     BNE $7FAE
7FB6 68        PLA
7FB7 AA        TAX
7FB8 68        PLA
7FB9 A8        TAY
7FBA 60        RTS
```

程序 No. 5－9 比 No. 5－6 少占用 162 个单元，程序量压缩了 46.6%。可见，灵活的调用系统子程序可使程序得到最大限度的优化。

为使初入门的朋友也能使用简化的电子钟程序，特把 No. 5－9 的 F BASIC 写入、运行程序列于 No. 5－10。

```
5 REM No. 5--10 电子钟简化写入、运行程序
10 FOR I=0 TO 185 : READ A$
20 POKE &7F01 + I, VAL( "&" + A$ )
30 NEXT
40 DATA A9, 0A, 85, EE, A9, 7F, 85, EF, 60, 48,20, 12, 7F, 68, 4C, 0F, 88, EE, F0, 7F,AD, F0,
   7F, C9, 32, F0, 01, 60, A9, 00,8D, F0, 7F, EE, F1, 7F, AD, F1, 7F, C9, 3C, D0, 23, A9, 00, 8D,
   F1, 7F, EE, F2, 7F, AD, F2, 7F, C9, 3C, D0, 14, A9, 00, 8D, F2, 7F
50 DATA EE, F3, 7F, AD, F3, 7F, C9, 0D, 90, 05, A9, 00, 8D, F3, 7F, 98, 48, 8A, 48, A2, 1F, B5,
   28, 48, CA, 10, FA, A0, 00, A9, 02, 8D, F4, 7F, AE, F4, 7F, BD, F1, 7F, 85, 28, A9, 00, 85,
   29, 20, 3C, 8D, A5, 40, 99, F5, 7F, C8, A5, 41, 99, F5, 7F, C8, A9, 3A, 99, F5, 7F, C8, CE, F4,
   7F, AE, F4, 7F, 10, DA
60 DATA A5, 93, F0, 1D, 20, F1, AB, A0, 00, A9, 20, A2, 36, 20, F8, 81, B9, F5, 7F, 8D, 07, 20,
   C8, C0, 08, D0, F5, 20, FD, AB, 20, E6, AB, A2, 00, 68, 95, 28, E8, E0, 20, D0, F8, 68, AA,
   68, A8, 60
70 CALL 32513 : POKE 147, 1 : CLS                  （启动电子钟）
80 INPUT "INPUT <hour> "; A : POKE 32755,A         （调时）
90 INPUT "INPUT <minute>";A : POKE 32754,A         （调分）
100 INPUT "INPUT <second>";A : POKE 32753,A        （调秒）
110 CLS : LOCATE 5,5 : PRINT "WELCOME USING";: LOCATE 4, 7: PRINT"ELECTRONIC CLOCK"
120 PAUSE 300: CLS : NEW
```

5·3　局部画面的显示编程技巧

在第二章我们曾讨论了背景显示系统的操作原理，并介绍了背景画面显示、位移的基本编程方法；在第三章详细讨论了查表法绘图的典型示范程序 No. 3－4。下面将在以上讨论的基础上，较全面的介绍任天堂游戏中常用的背景画面绘图方法和编程技巧，并给出具体的、独立的应用子程序，供编程中调用。

局部画面显示是任天堂游戏中最常用的一种显示技术，如游戏中角色的生命值、人数、关

次的提示，游戏得分的累积，游戏进程的标志，角色对话的送显等都是通过局部送显完成的。一般这种形式的送显过程都是在画面的正常显示下进行的，因此不能采取关显示的方法，只能在中断期间把显示内容送入 PPU 的显示单元，本节将通过实例具体介绍几种常用的局部绘图方法。

5·3·1 直接送显法

例一、在屏幕的左上角(从 PPU 地址 $ 2005 开始)显示数字“1000”。

实现程序见 No. 5－11

No. 5－11 局部画面送显子程序（一）

```
7EF1  A9 FA     LDA # $FA
7EF3  85 EE     STA $EE
7EF5  A9 7E     LDA # $7E
7EF7  85 EF     STA $EF
7EF9  60        RTS
7EFA  20 00 7F  JSR $7F00
7EFD  4C 0F 88  JMP $880F
7F00  A9 20     LDA # $20
7F02  8D 06 20  STA $2006
7F05  A9 05     LDA # $05
7F07  8D 06 20  STA $2006
7F0A  A9 31     LDA # $31
7F0C  8D 07 20  STA $2007
7F0F  A9 30     LDA # $30
7F11  8D 07 20  STA $2007
7F14  8D 07 20  STA $2007
7F17  8D 07 20  STA $2007
7F1A  60        RTS
```

只要把 No. 5－11 放入中断处理程序中，即可送显例一数据。写入中断程序的方法已在 5－1 节详细讨论，读者可参照编写 F BASIC 语言的写入内存程序，写入完毕只要执行“CALL &7EF1”，就可完成送显。(注——在以后的讨论中，为压缩篇幅，除特殊情况外，一般不再列举 F BASIC 的写入程序。)

但是，在实际应用中，送显的数据往往是变化的，由于我们不可能在程序运行时反复改变中断程序，所以 No. 5－11 仅仅适用于送显数据不变的场合。当送显诸如积分类变化的数字时，应采用例二的程序 No. 5－12。

例二、在屏幕的右上角(从 PPU 地址 $ 2015 开始)显示游戏者的四位即时积分。

我们选取 $ A0～ $ A3 四个连续的零页单元依次存放个、十、百、千四位数据，实现程序见 No. 5－12。

No. 5－12 局部画面送显子程序(二)

```
7EF1  A9 FA     LDA # $FA
7EF3  85 EE     STA $EE
7EF5  A9 7E     LDA # $7E
7EF7  85 EF     STA $EF
```

```
7EF9  60        RTS
7EFA  20 00 7F  JSR $7F00
7EFD  4C 0F 88  JMP $880F
7F00  8A        TXA
7F01  48        PHA
7F02  A9 20     LDA #$20
7F04  A2 15     LDX #$15
7F06  20 F8 81  JSR $81F8
7F09  A2 03     LDX #$03
7F0B  B5 A0     LDA $A0,X
7F0D  8D 07 20  STA $2007
7F10  CA        DEX
7F11  10 F8     BPL $7F0B
7F13  68        PLA
7F14  AA        TAX
7F15  60        RTS
```

运行程序 No. 5－12 后，只要向 $A0～$A3 单元置入数据，就可立即显示。当欲在画面上连续显示多个内容时，只要选取相应的连续单元数，修改 $7F09 行 X 的初值即可。但是，当显示的内容较多时，不可能有这么多连续的零页单元供我们使用，所以还是采用查表法为宜。

5・3・2 查表送显法

例三、使用查表法局部送显图 5－1 的内容。

2000	TIME	SCORE	STAGE	LEVEL
2020	08:35	01234	3	8

图 5－1 局部送显内容

查表法局部送显程序见 No. 5－13。

No. 5－13 局部送显子程序(三)

```
7EF1  A9 FA     LDA #$FA
7EF3  85 EE     STA $EE
7EF5  A9 7E     LDA #$7E
7EF7  85 EF     STA $EF
7EF9  60        RTS
7EFA  20 00 7F  JSR $7F00
7EFD  4C 0F 88  JMP $880F
7F00  8A        TXA
7F01  48        PHA
7F02  A9 20     LDA #$20
7F04  A2 01     LDX #$01
7F06  20 F8 81  JSR $81F8
7F09  A2 00     LDX #$00
```

```
7F0B  BD 21 7F   LDA $7F21,X
7F0E  F0 06      BEQ $7F16
7F10  8D 07 20   STA $2007
7F13  E8         INX
7F14  10 F5      BPL $7F0B
7F16  68         PLA
7F17  AA         TAX
7F18  60         RTS
7F20     54 49 4D 45 20 20 20 20 53 43 4F 52 45 20 20
         T  I  M  E              S  C  O  R  E
7F30  20 20 53 54 41 47 45 20 20 20 4C 45 56 45 4C 20
            S  T  A  G  E           L  E  V  E  L
7F40  20 30 38 3A 33 35 20 20 20 30 31 32 33 34 20 20
         0  8  :  3  5           0  1  2  3  4
7F50  20 20 20 20 33 20 20 20 20 20 20 20 38 00
                  3                       8
```

程序 No. 5－13 是以读出为零的数据为标志而结束送数的，它不仅能够进行局部绘图，而且通过简单的扩充数据区就能够绘出整幅图形，因此，程序 No. 5－13 是一个实用性极强的局部绘图程序。

5·3·3　开关变址送显法

以上介绍的绘图子程序都必须安装在中断处理程序中运行，也就是说：每执行一次中断，就要把它们的内容送显一次。实际上大多数送显内容不可能变化的那么快，如时间的显示，一秒钟才改变一次，那么在一秒中的五十次中断处理中有四十九次的送显是重复的，这无疑增大了中断处理时间的开销。若中断处理程序较短小则影响不大，但当中断程序量较大时，就有可能造成一个场回扫时间不够的情况，从而造成显示紊乱。因此，一般在中断处理的各子程序开始设置一个执行软开关，只有开关接通时才执行该子程序，当开关断开时则跳过该子程序执行其它中断程序。这样处理可减少中断执行时间的开销，加快中断执行速度，提高中断工作效率。

另外在实际局部送显中，常有大部分内容不变，少部分内容改变的情况，如图 5－1 中的第一行是不变的，只要送显一次就可保持，不必重复送显；只有第二行是变化的。那么在应用中，对第一行只要送显一次就足够了，以后的任务是送显第二行的可变部分。所以，要求送显子程序中 PPU 的地址是可变的。只有这样才能使送显子程序的使用更灵活，通用性更强。

例四、使用开关变址送显法送显图 5－1 的内容。

送显程序见 No. 5－14

No. 5－14 局部送显子程序(四)

```
7EF1  A9 FA      LDA #$FA
7EF3  85 EE      STA $EE
7EF5  A9 7E      LDA #$7E
7EF7  85 EF      STA $EF
7EF9  60         RTS
7EFA  20 00 7F   JSR $7F00
7EFD  4C 0F 88   JMP $880F
```

```
7F00  AD FF 04   LDA  $04FF      ;以$4FF为执行软开关
7F03  D0 01      BNE  $7F06      ;$4FF≠0执行送显
7F05  60         RTS             ;$4FF=0则直接返回
7F06  8A         TXA
7F07  48         PHA
7F08  AD 30 7F   LDA  $7F30
7F0B  AE 31 7F   LDX  $7F31
7F0E  20 F8 81   JSR  $81F8      ;置PPU地址
7F11  A2 00      LDX  #$00
7F13  BD 32 7F   LDA  $7F32,X
7F16  F0 06      BEQ  $7F1E
7F18  8D 07 20   STA  $2007
7F1B  E8         INX
7F1C  D0 F5      BNE  $7F13
7F1E  20 E6 AB   JSR  $ABE6      ;写PPU后的屏幕定位
7F21  A9 00      LDA  #$00
7F23  8D FF 04   STA  $04FF      ;关闭送显开关
7F26  68         PLA
7F27  AA         TAX
7F28  60         RTS
```

当程序 No. 5－14 装入中断处理程序后，运行 F BASIC 程序 No. 5－15 即可送显图 5－1 的局部画面。

```
5 REM No. 5－15 局部送显子程序(四)的辅助程序
10 POKE &7F30,20,01,54,49,4D,45,20,20,20,20,
             (PPU首址T  I  M  E)
    53,43,4F,52,45,20,20,20,20,53,54,41,47,
  (  S  C  O  R  E              S  T  A  G)
    45,20,20,20,4C,45,56,45,4C,00
  (  E           L  E  V  E  L  送显结束标志)
20 POKE &4FF,1
30 POKE &7F30,20,21,30,38,3A,33,35,20,20,20,
         (PPU首址   0  8  :  3  5)
    30,31,32,33,34,20,20,20,20,20,20,33,20,
  (  0  1  2  3  4                    3)
    20,20,20,20,20,20,38,00
  (                     8)
40 POKE &04FF,1
50 END
```

注——括号中的内容为注释，不要输入程序中。

显而易见，使用程序 No. 5－14、No. 5－15 可任意向屏幕的任意位置送显任意内容。

5·3·4 F BASIC 系统送显法

当局部送显的画面位于 F BASIC 的显示画面中时，还有更简单的编程方法。

例五、在 F BASIC 画面座标(7, 5)处送显“WELCOME USING”,在(6,7)处送显“FAMILY COMPUTER”。

实现程序见 No. 5－16

No. 5－16 局部送显子程序(五)

```
7F00  A9 00      LDA  # $00
7F02  85 44      STA  $44
7F04  A5 44      LDA  $44
7F06  C9 02      CMP  # $02
7F08  F0 07      BEQ  $7F11
7F0A  20 12 7F   JSR  $7F12      ;赋值子程序
7F0D  E6 44      INC  $44
7F0F  D0 F3      BNE  $7F04
7F11  60         RTS             ;送显结束返回
    *  *    *    *
7F12  0A         ASL
7F13  0A         ASL
7F14  AA         TAX
7F15  BD 00 7E   LDA  $7E00,X
7F18  85 48      STA  $48        ;取 X 座标
7F1A  BD 01 7E   LDA  $7E01,X
7F1D  85 49      STA  $49        ;取 Y 座标
7F1F  BD 02 7E   LDA  $7E02,X
7F22  85 52      STA  $52        ;取数据区首址
7F24  BD 03 7E   LDA  $7E03,X
7F27  85 53      STA  $53
7F29  4C F8 87   JMP  $87F8      ;转读数送显子程序,送显结束后返回
7E00  07    05    00    7D    06    07    0F    7D
      座标      取数首址      座标      取数首址
7D00  57 45 4C 43 4F 4D 45 20 55 53 49 4E 47 00
       W  E  L  C  O  M  E     U  S  I  N  G
7D0F  46 41 4D 49 4C 59 20 20 43 4F 4D 50 55 54 45 52 00
       F  A  M  I  L  Y        C  O  M  P  U  T  E  R
```

程序 No. 5－16 由于调用了系统子程序,故短小精悍,玲珑剔透。显然,只要调整 $ 7E00 数据区就可随意控制显示位置和显示内容。

以上介绍的五个局部送显子程序中,前四个子程序都要写入中断程序才能运行,而子程序(五)—No. 5－16 则可独立运行,这是它的突出特点。

5・3・5　F BASIC 程序快速送显法

使用 BS. 2A 版本 F BASIC 的朋友,还可以不编机器语言程序,直接编写 F BASIC 语言程序快速送显局部画面。

例六、编写 F BASIC 语言程序快速送显背景图形库中第 32～254 号图形。

实现程序见 No. 5－17

```
5 REM No. 5－17 局部送显子程序(六)
```

```
10 CLS
20 FOR I=0 TO 223 : POKE 1792+I, I+32: NEXT
30 POKE 98,225
40 CALL 34811
```

程序 No. 5－17 中的 30 行写入 98 单元的数据为一次送显的数据个数加一，调整该数值可以控制一次送显的字符、图形数量。我们还可以在 40 行之前加一光标定位语句"LOCATE X,Y" 用于控制送显内容的初始座标。由于该子程序调用了系统送显子程序 $87FB,因而送显速度极快。有兴趣的朋友可以与用"PRINT CHR $ "语句送显的方法比较一下程序运行速度,可以发现 No. 5－17 要快许多倍。

现在我们再使用 No. 5－17 试验送显例五的内容。程序见 No. 5－18

```
5 REM No. 5-18 局部送显子程序(七)
10 POKE 1792,&57,&45,&4C,&43,&4F,&4D,&45,&20,&55,&53,&49,&4E,&47
20 POKE 98, 15 : LOCATE 7, 5: CALL 34811
30 POKE 1792,&46,&41,&4D,&49,&4C,&59,&20,&20,&43,&4F,&4D,&50,&55,&54,&45,&52
40 POKE 98, 17 : LOCATE 6,7 :CALL 34811
```

运行 No. 5－18 可见,它达到了 No. 5－16 同样的效果,程序 No. 5－18 再一次体现了灵活调用系统子程序的优越性,同时也表明,不编写机器语言子程序也能加快程序的运行速度。

5・4 背景画面绘制的编程技巧

在背景画面的绘制中,常采取关显示后查表送显的方法。查表法绘图有多种形式,本节仅选择介绍几种最常用的方法。

5・4・1 直接送显绘图法(《小学算术一年级》绘图模式)

《小学算术一年级》游戏是日本东京小学推荐的优秀教学软件,它的背景画面绘制采取了最原始的方法——直接查表法。其绘图源程序如 No. 5－19。

No. 5－19《小学算术一年级》绘图源程序

```
AD04  8518    STA $18        ;保护 A 值
AD06  A900    LDA # $00
AD08  8516    STA $16        ;设置取数方式
AD0A  8619    STX $19        ;保护 X 值
AD0C  841A    STY $1A        ;保护 Y 值
AD0E  68      PLA
AD0F  850D    STA $0D
AD11  68      PLA
AD12  850E    STA $0E
AD14  A001    LDY # $01
AD16  B10D    LDA ($0D),Y
AD18  8511    STA $11        ;绘图数据区地址低位
AD1A  C8      INY
AD1B  B10D    LDA ($0D),Y
AD1D  8512    STA $12        ;绘图数据区地址高位
```

```
AD1F  A50D     LDA $0D
AD21  18       CLC
AD22  6902     ADC #$02
AD24  850D     STA $0D
AD26  9002     BCC $AD2A
AD28  E60E     INC $0E
AD2A  A50E     LDA $0E
AD2C  48       PHA
AD2D  A50D     LDA $0D
AD2F  48       PHA              ;存子程序返回地址
AD30  A000     LDY #$00
AD32  A516     LDA $16
AD34  C902     CMP #$02
AD36  F00A     BEQ $AD42        ;若$16=2 则不改变 PPU 地址
AD38  B111     LDA ($11),Y
AD3A  850F     STA $0F          ;PPU 地址低位
AD3C  C8       INY
AD3D  B111     LDA ($11),Y
AD3F  8510     STA $10          ;PPU 地址高位
AD41  C8       INY
AD42  B111     LDA ($11),Y
AD44  8514     STA $14          ;每组送数的数量
AD46  C8       INY
AD47  B111     LDA ($11),Y
AD49  8515     STA $15          ;本次送数组数
AD4B  C8       INY
AD4C  A514     LDA $14
AD4E  C515     CMP $15
AD50  9071     BCC $ADC3        ;若$14<$15 则转$ADC3,
AD52  A615     LDX $15          ;改为纵向送数
AD54  8A       TXA
AD55  48       PHA              ;送数组数入栈保护
AD56  A504     LDA $04
AD58  D032     BNE $AD8C
AD5A  A500     LDA $00
AD5C  48       PHA
AD5D  2035AC   JSR $AC35        ;屏蔽$2000 的 D7 位
AD60  A500     LDA $00
AD62  8D0020   STA $2000        ;设置送数方式
AD65  A510     LDA $10
AD67  8D0620   STA $2006        ;置 PPU 地址高位
AD6A  A50F     LDA $0F
AD6C  8D0620   STA $2006        ;置 PPU 地址低位
AD6F  A614     LDX $14          ;以 X 为取数计数器
```

```
AD71  B111     LDA ($11),Y
AD73  8D0720   STA $2007       ;从数据区取数送入 PPU
AD76  A516     LDA $16
AD78  C901     CMP #$01        ;若$16=01 则一个数据连送 X 次
AD7A  F006     BEQ $AD82       ;若$16=00 则连送其后的 X 个数据
AD7C  E611     INC $11         ;令取数地址低位加一
AD7E  D002     BNE $AD82
AD80  E612     INC $12         ;若低位为零则令高位加一
AD82  CA       DEX
AD83  D0EC     BNE $AD71
AD85  68       PLA
AD86  2039AC   JSR $AC39
AD89  4CACAD   JMP $ADAC
AD8C  205DAE   JSR $AE5D
AD8F  A614     LDX $14
AD91  8A       TXA
AD92  2081AE   JSR $AE81
AD95  B111     LDA ($11,Y)
AD97  2081AE   JSR $AE81
AD9A  A516     LDA $16
AD9C  C901     CMP #$01
AD9E  F006     BEQ $ADA6
ADA0  E611     INC $11
ADA2  D002     BNE $ADA6
ADA4  E612     INC $12
ADA6  CA       DEX
ADA7  D0EC     BNE $AD95
ADA9  208DAE   JSR $AE8D
ADAC  A50F     LDA $0F         ;令 PPU 地址加 H20,使其指向下一行
ADAE  18       CLC
ADAF  6920     ADC #$20
ADB1  850F     STA $0F
ADB3  9002     BCC $ADB7
ADB5  E610     INC $10         ;若有进位则令高位加一
ADB7  68       PLA
ADB8  AA       TAX
ADB9  CA       DEX             ;令送数组数计数器减一
ADBA  D098     BNE $AD54       ;不为零则转$AD54 继续送数
ADBC  A518     LDA $18         ;若为零则恢复原 A、X、Y 寄存器值,
ADBE  A619     LDX $19         ;结束送数
ADC0  A41A     LDY $1A
ADC2  60       RTS
```

复位画面数据区如下：

```
B45E                                   00 20
```

B460 20 20 CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B461:每组送数数量置于$14。以后的数据都是绘图数据
$B460:本次送数的组数存于$15
B470 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B480 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B490 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B4A0 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B4B0 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B4C0 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B4D0 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B4E0 CE CE CE CE CE CE CE CE CE CE E0 E1 E2 E3 82 83
B4F0 82 83 CE CE CE A2 A3 CE CC CD CE CE CE CE CE CE
B500 CE CE CE CE CE CE CE CE CE CE E4 E5 E6 E7 92 93
B510 92 93 CE CE CE B2 B3 CE DC DD CE CE CE CE CE CE
B520 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B530 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B540 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B550 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B560 CE CE CE CE CE CE CE CE CE CE CE 80 81 80 81 E0
B570 E1 E2 E3 E8 E9 60 71 CC CD CE CE CE CE CE CE CE
B580 CE CE CE CE CE CE CE CE CE CE CE 90 91 90 91 E4
B590 E5 E6 E7 EA EB CE CE DC DD CE CE CE CE CE CE CE
B5A0 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B5B0 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B5C0 CE CE CE CE CE CE C4 C6 C6 C6 C6 C6 C6 C6 C6 C6
B5D0 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C5 CE CE
B5E0 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF 8C 8D
B5F0 C8 CF E0 19 E2 E3 CF CF CF CF CF CF CF D7 CE CE
B600 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF 9C 9D
B610 D8 D9 E4 E5 E6 E7 CF CF CF CF CF CF CF D7 CE CE
B620 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF
B630 CF CF CF CF CF CF CF CF CF CF CF CF CF D7 CE CE
B640 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF 84 85
B650 CA CB E0 19 E2 E3 CF CF CF CF CF CF CF D7 CE CE
B660 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF 94 95
B670 DA DB E4 E5 E6 E7 CF CF CF CF CF CF CF D7 CE CE
B680 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF
B690 CF CF CF CF CF CF CF CF CF CF CF CF CF D7 CE CE
B6A0 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF 8C 8D
B6B0 C8 CF E0 19 E2 E3 CF CF CF CF CF CF CF D7 CE CE
B6C0 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF 9C 9D
B6D0 D8 D9 E4 E5 E6 E7 CF CF CF CF CF CF CF D7 CE CE
B6E0 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF

```
B6F0  CF CF CF CF CF CF CF CF CF CF CF CF CF D7 CE CE
B700  CE CE CE CE CE CE C7 CF CF CF CF CF CF CF 84 85
B710  CA CB E0 19 E2 E3 CF CF CF CF CF CF CF D7 CE CE
B720  CE CE CE CE CE CE C7 CF CF CF CF CF CF CF 94 95
B730  DA DB E4 E5 E6 E7 CF CF CF CF CF CF CF D7 CE CE
B740  CE CE CE CE CE CE D4 D6 D6 D6 D6 D6 D6 D6 D6 D6
B750  D6 D6 D6 D6 D6 D6 D6 D6 D6 D6 D6 D6 D6 D5 CE CE
B760  CE CE CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B770  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B780  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B790  CF 08 18 09 08 CF CF 09 CF 18 CF CF CF CF CF CF
B7A0  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B7B0  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B7C0  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B7D0  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B7E0  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B7F0  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B800  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B810  CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B820  CF CF 00 00 00 00 00 00 00 00 00 00 00 0B 02 08
B830  00 00 50 50 5E 52 50 5F 50 50 55 55 55 55 55 55
B840  55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
B850  55 55 00 00 00 00 00 00 00 00 50 50 50 50 50 50
B860  50 50
```

No. 5－19 所调用的子程序

```
AC35  A500      LDA  $00
AC37  297F      AND  #$7F
AC39  8D0020    STA  $2000
AC3C  8500      STA  $00
AC3E  60        RTS
```

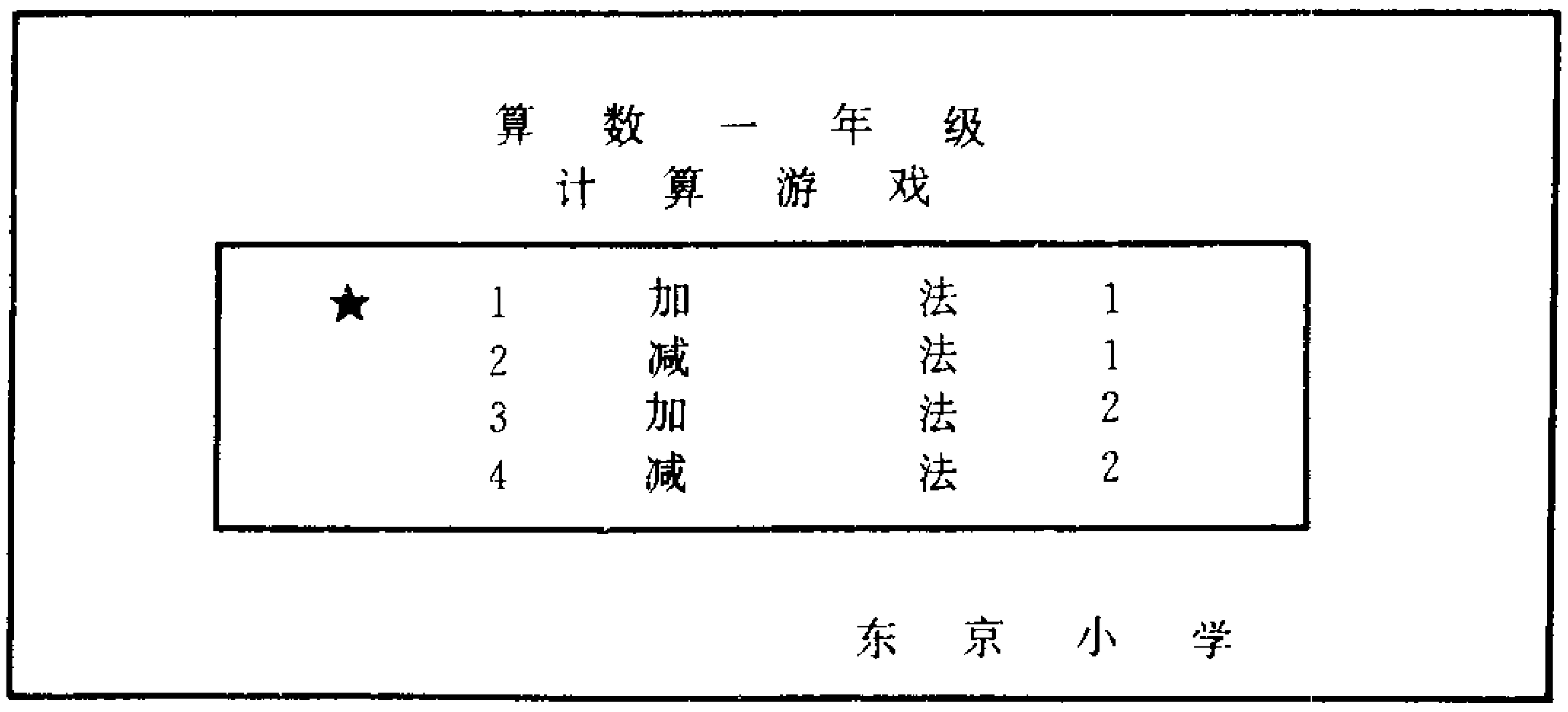
程序 No. 5－19 的绘图数据如图 5－2。

列次	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	行次
2000	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	00
2020	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	01
2040	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	02
2060	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	03
2080	CE	CE	CE	CE	CE	CE	CE	CE	E0	E1	E2	E3	82	83	82	83	CE	CE	CE	A2	A3	CE	CC	CD	CE	CE	CE	CE	CE	CE	CE	CE	04
20A0	CE	CF	CE	CE	CE	CE	CE	CE	E4	E5	E6	E7	92	93	92	93	CE	CE	CF	B2	B3	CE	DC	DD	CE	CE	CF	CE	CE	CE	CE	CE	05
20C0	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CF	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	06
20E0	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CF	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	07
2100	CE	CE	CE	CE	CE	CE	CE	CE	CE	80	81	80	81	E0	E1	E2	E3	E8	E9	60	71	CC	CD	CE	CE	CE	CE	CE	CE	CE	CE	CE	08
2120	CE	CE	CE	CE	CE	CE	CE	CE	CE	90	91	90	91	E4	E5	E6	E7	EA	EB	CE	CE	DC	DC	CE	CE	CE	CE	CE	CE	CE	CE	CE	09
2140	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	CE	0A
2160	CE	CE	CE	CE	C4	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C6	C5	CE	CE	CE	CE	0B
2180	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	8C	8D	C8	·	E0	19	E2	E3	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	0C
21A0	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	9C	9D	D8	D9	E4	E5	E6	E7	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	0D
21C0	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	0E
21E0	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	84	85	CA	CB	E0	19	E2	E3	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	0F
2200	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	94	95	DA	DB	E4	E5	E6	E7	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	10
2220	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	11
2240	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	8C	8D	C8	·	E0	19	E2	E3	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	12
2260	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	9C	9D	D8	D9	E4	E5	E6	E7	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	13
2280	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	14
22A0	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	84	85	CA	CB	E0	19	E2	E3	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	15
22C0	CE	CE	CE	CE	C7	·	·	·	·	·	·	·	94	95	DA	DB	E4	E5	E6	E7	·	·	·	·	·	·	·	D7	CE	CE	CE	CE	16
22E0	CE	CE	CE	CE	D4	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D5	D6	D6	D6	D5	CE	CE	CE	CE	17
2300	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	18
2320	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	08	18	09	08	·	·	09	·	18	·	·	·	·	·	·	·	·	19
2340	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1A
2360	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1B
2380	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1C
23A0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1D
23C0	00	00	00	00	00	00	00	50	00	00	00	0B	02	08	00	00	50	50	5E	32	50	5F	50	50	55	55	55	55	55	55	55	55	
23E0	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	00	00	00	00	00	00	00	00	50	50	50	50	50	50	50	50	

注——“·”代表 CFH； ￥23C0～￥23FF 两行为画面配色数据。

图 5—2 《小学算数一年级》复位画面

绘图效果如图 5－3。



注——实际绘图中的汉字是日文

图 5－3 《小学算术一年级》复位画面

程序 No. 5－19 中已作了详细注释，不多解释。下面给出在 BS. 2A 系统中可直接调用的子程序 No. 5－20。

No. 5－20(直接送显)背景绘图子程序(一)

```
7EF1  A9 FA     LDA #$FA
7EF3  85 EE     STA $EE
7EF5  A9 7E     LDA #$7E
7EF7  85 EF     STA $EF
7EF9  60        RTS
7EFA  20 00 7F  JSR $7F00
7EFD  4C 0F 88  JMP $880F
      *    *    *    *    *
7F00  48        PHA
7F01  8A        TXA
7F02  48        PHA
7F03  98        TYA
7F04  48        PHA
7F05  A9 00     LDA #$00
7F07  85 50     STA $50        ;绘图数据区地址低位存$50
7F09  A9 78     LDA #$78
7F0B  85 51     STA $51        ;绘图数据区地址高位存$51
7F0D  A0 00     LDY #$00
7F0F  B1 50     LDA ($50),Y
7F11  85 52     STA $52        ;PPU 地址低位存$52
7F13  C8        INY
7F14  B1 50     LDA ($50),Y
7F16  85 53     STA $53        ;PPU 地址高位存$53
7F18  C8        INY
7F19  B1 50     LDA ($50),Y
7F1B  85 54     STA $54        ;每组送数的数量存$54
7F1D  C8        INY
7F1E  B1 50     LDA ($50),Y
```

```
7F20  85 55     STA $55        ;本次送数组数存$55
7F22  C8        INY
7F24  A6 55     LDX $55
7F26  8A        TXA
7F27  48        PHA            ;送数组数入栈保护
7F28  A5 32     LDA $32
7F2A  48        PHA            ;$2000原值入栈
7F2B  20 67 7F  JSR $7F67      ;屏蔽$2000的D7位
7F2E  A5 32     LDA $32
7F30  8D 00 20  STA $2000      ;设置送数方式
7F33  A5 53     LDA $53
7F35  8D 06 20  STA $2006      ;置PPU地址高位
7F38  A5 52     LDA $52
7F3A  8D 06 20  STA $2006      ;置PPU地址低位
7F3D  A6 54     LDX $54        ;以X为取数计数器
7F3F  B1 50     LDA ($50),Y
7F41  8D 07 20  STA $2007      ;从数据区取数送入PPU
7F44  E6 50     INC $50        ;令取数地址低位加一
7F46  D0 02     BNE $7F4A
7F48  E6 51     INC $51        ;若低位为零则令高位加一
7F4A  CA        DEX
7F4B  D0 F2     BNE $7F3F
7F4D  68        PLA
7F4E  20 6B 7F  JSR $7F6B
7F51  A5 52     LDA $52        ;令PPU地址加H20,使其指向下一行
7F53  18        CLC
7F54  69 20     ADC #$20
7F56  85 52     STA $52
7F58  90 02     BCC $7F5C
7F5A  E6 53     INC $53        ;若有进位则令高位加一
7F5C  68        PLA
7F5D  AA        TAX
7F5E  CA        DEX            ;令送数组数计数器减一
7F5F  D0 C5     BNE $7F26      ;不为零则转$7F26继续送数
7F61  68        PLA            ;若为零则恢复原A、X、Y寄存器值,
7F62  AA        TAX            ;结束送数
7F63  68        PHA
7F64  A8        TAY
7F65  68        PLA
7F66  60        RTS
      *    *    *    *    *
7F67  A5 32     LDA $32
7F69  29 7F     AND #$7F
7F6B  8D 00 20  STA $2000
```

```
7F6E  85 32       STA $32
7F70  60          RTS
```

现在,只要把绘图数据按 No. 5－19 中＄B45E 的格式写入＄7800,就可绘出你希望的背景画面。限于篇幅,不再举例,留给读者练习。

从 No. 5－19、No. 5－20 可明显看出,直接送显绘图法的优点是:绘图数据区为平面结构,数据与显示单元的关系一一对应,单纯从数据区就可看出绘图效果,背景画面设计直观、容易,修改方便,擅长于细腻画面的描绘,特别适合初学者使用;缺点是:由于每一个显示单元和配色单元都要占一个内存单元,一幅背景画面要占用 1024 个内存单元,故数据区较庞大。

5・4・2　BS. 2A 背景绘图法(《大金刚》游戏绘图)

在第三章我们曾对系统背景绘图子程序 No. 3－4 进行了详细的讨论,本节仅以《大金刚》一代游戏的第二关背景图形的绘制为例,介绍系统绘图子程序的应用。(因《大金刚》一代第二关的背景图形字模与我们的电脑系统的背景字模有些相似,故模拟比较容易。)

画面的绘图效果见《大金刚》一代第二关,在此仅将背景画面的设计中各显示单元的图形代号填入图 5－4(代号以拙作《电脑游戏机 F BASIC 语言与编程技巧》中的背景图形库的标识为准)。

对照图 5－4,我们只要按程序 No ・3－4 的送数指南格式建立数据区后,再编写一个调用程序 No. 3－4 的子程序就可以绘出《大金刚》一代第二关的游戏画面。画面数据如下:

```
7ED0  20 24 04 09 11 09 11 20 44 04 C4 1C 1C C6 20 63
7EE0  06 CF 0C BF BF 18 CF 20 83 06 E0 D8 C2 C2 DA E1
7EF0  20 A4 04 1B C2 C2 04 20 C3 0A C9 BB 20 20 BB C9
7F00  20 D2 20 D2 20 0A C5 D2 20 0C C5 D2 20 8D 46 D0
7F10  20 AA 43 D0 20 B2 C2 D2 20 E1 10 D0 D0 D0 D0 C4
7F20  C5 C6 D0 D0 D0 D0 D0 D0 C4 C5 C6 21 F1 47 D0 21
7F30  25 03 C4 C5 C6 21 2E 03 C4 C5 C6 21 06 95 D1 C5
7F40  D3 D3 CF D3 D3 CF D3 D3 CF D3 D3 CF D3 D3 CF D3
7F50  D3 C5 D1 21 0F 95 D1 C5 D3 D3 CF D3 D3 CF D3 D3
7F60  CF D3 D3 CF D3 D3 CF D3 D3 C5 D1 22 02 43 D0 21
7F70  E9 44 D0 22 24 C5 D2 22 0A C7 D2 22 0C C7 D2 22
7F80  E3 C4 D2 21 17 C3 D2 21 5D C4 D2 21 B5 C3 D2 21
7F90  FB C3 D2 21 3D 42 D0 21 5A 42 D0 21 77 42 D0 21
7FA0  93 43 D0 21 DB 45 D0 22 15 42 D0 22 38 42 D0 22
7FB0  5C 42 D0 22 7E 42 D0 22 9E 01 D2 22 BD 43 D0 22
7FC0  C2 43 D0 22 EA 43 D0 22 FA 42 D0 23 38 42 D0 23
7FD0  62 06 D0 D0 D0 C4 C5 C6 23 6E 08 C4 C5 C6 20 20
7FE0  D0 D0 D0 23 A0 45 D0 23 A5 03 C8 CA C9 23 A8 46
7FF0  D0 23 AE 03 C8 CA C9 23 B1 4F D0 00
```

列次	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
2000											H3		H3																			
2020					B2	C2	B2	C2			H3		H3																			
2040					F5	D5	D5	F7			H3		H3																			
2060				G8	B5	E8	E8	D1	G8		H3		H3																			
2080				J1	I1	F3	F3	I3	J2		H3		H3	H1	H1	H1	H1	H1	H1													
20A0					D4	F3	F3	A5			H1	H1	H1						H3													
20C0				G2	E4				G2		H3		H3						H3													
20E0		H1	H1	H1	H1	F5	F6	F7	H1	H1	H1	H1	H1	H1	F5	F6	F7	H1	H1	H1	H1	H1	H1	H1								
2100							H2									H2								H3								
2120						F5	F6	F7							F5	F6	F7							H3						H1	H1	
2140							H4									H4								H3			H1	H1		H3		
2160							H4									H4								H1	H1					H3		
2180							G8									G8				H1	H1	H1								H3		
21A0							H4									H4						H3								H3		
21C0							H4									H4						H3						H1	H1	H1	H1	H1
21E0							G8			H1	H1	H1	H1			G8						H3						H3				
2200			H1	H1	H1		H4				H3		H3			H4						H1	H1					H3				
2220					H3		H4				H3		H3			H4									H1	H1		H3				
2240					H3		G8				H3					G8												H1	H1			
2260					H3		H4				H3					H4															H1	H1
2280					H3		H4				H3					H4															H3	
22A0					H3		G8				H3					G8														H1	H1	H1
22C0			H1	H1	H1		H4				H3					H4																
22E0				H3			H4				H1					H4											H1	H1				
2300				H3			G8									G8																
2320				H3			H4									H4								H1	H1							
2340				H3			H4									H4																
2360			H1	H1	H1	F5	F6	F7							F5	F6	F7			H1	H1	H1										
2380							H2									H2																
23A0	H1	H1	H1	H1	H1	G1	G3	G2	H1	H1	H1	H1	H1	H1	G1	G3	G2	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1

图5－4 《大金刚》一代第二关画面的图形代号

以上数据区仅占用了300个内存单元，比直接送显绘图法要节约三分之二的数据区，这是它的突出优势。

使用BS.2A版本F BASIC的朋友可直接调用系统送显子程序，调用程序见No.5－21。

No.5－21(BS.2A绘图子程序$EE43调用的子程序)

背景绘图子程序(二)

```
7EA0  48        PHA
7EA1  8A        TXA
7EA2  48        PHA
7EA3  98        TYA
7EA4  48        PHA
7EA5  A5 00     LDA $00
7EA7  48        PHA
7EA8  A5 01     LDA $01
7EAA  48        PHA        ;保护现场
7EAB  A9 00     LDA #$00
7EAD  8D 01 20  STA $2001  ;关显示
7EB0  A9 D0     LDA #$D0
7EB2  85 00     STA $00
7EB4  A9 7E     LDA #$7E
7EB6  85 01     STA $01    ;向$00、$01单元置入数据区首址
7EB8  20 43 EE  JSR $EE43  ;系统送显子程序
7EBB  A9 1E     LDA #$1E
7EBD  8D 01 20  STA $2001  ;开显示
7EC0  68        PLA
7EC1  85 01     STA $01
7EC3  68        PLA
7EC4  85 00     STA $00
7EC6  68        PLA
7EC7  A8        TAY
7EC8  68        PLA
7EC9  AA        TAX
7ECA  68        PLA
7ECB  60        RTS
```

当欲在其它F BASIC版本下运行No.5－21子程序时，则要将No.3－4合成到No.5－21中才能绘出背景图案。程序的合成留给读者练习。

5·4·3　置色绘图法(《大赛车》游戏画面绘制模式)

任天堂游戏《大赛车》复位画面的绘制方法与BS.2A的背景绘图法截然不同，它的标题画面大黑体美术字“F－1RACE”的绘制，采取了较为奇特的绘图方法，使程序量、数据区都得到了极大的压缩。现介绍给大家，《大赛车》的绘图源程序见No.5－22

No.5－22《大赛车》复位画面绘图子程序

```
E62B  A920      LDA #$20
E62D  A040      LDY #$40
```

```
E62F  2064E9   JSR $E964    ;置PPU送数首址$2040
E632  A000     LDY #$00
E634  B96DE8   LDA $E86D,Y
E637  F022     BEQ $E65B
E639  C920     CMP #$20
E63B  B018     BCS $E655
E63D  290F     AND #$0F
E63F  AA       TAX
E640  B96DE8   LDA $E86D,Y
E643  2910     AND #$10
E645  D004     BNE $E64B
E647  A92D     LDA #$2D
E649  D002     BNE $E64D
E64B  A92E     LDA #$2E
E64D  8D0720   STA $2007
E650  CA       DEX
E651  10FA     BPL $E64D
E653  3003     BMI $E658
E655  8D0720   STA $2007
E658  C8       INY
E659  D0D9     BNE $E634
E65B  60       RTS
```

绘图数据区

```
E86D                                                     0A 13 FE
E870  02 A9 11 09 2F 2D 2F 2D 2F 2D 2F 2D 2F 01 11 05
E880  12 01 2F 2D 2F 2D 2F 2D 2F 2D 2D 2F 2D 2F 2D 2F
E890  2D 2F 02 12 FE 2D 11 01 11 02 2F 2D 2F 2D 2F 2D
E8A0  2F 2F 2D 2F 2D 2F 2D 2F 03 11 06 11 03 2F 2D 2F
E8B0  2D 2F 2D 2D 2F 2D 2F 2D 2F 04 11 06 11 04 2F 2D
E8C0  2F 2D 2F 0F 0F 0F 0F 04 13 FB 2D FA 12 FB 2D FA
E8D0  12 FB 2D 13 FE 08 11 2D 11 2D 11 2D 11 2D 11 2D
E8E0  11 2D 11 0B 13 FD 2D 11 2D 11 2D 11 03 12 FE 09
E8F0  11 FF 2E BA 2D 14 2D 11 2D 11 2D 11 0B 11 2D FF
E900  2E 2D 11 2D 11 2D FC 12 FD 2D 13 FE 03 00 00 00
```

程序No.5－22中，若从数据区读出的数据为零则结束送数；若读出的数据大于或等于H20则将该数直接送入PPU；若读出的数据小于H20，且该数的D4位为零则以该数为计数器连向PPU送H2D(《大赛车》游戏中背景图形库的H2D图形为空格)；若读出的数据小于H20，且其D4位为1，则以该数的低四位为计数器连向PPU送H2D。程序No.5－22仅占用数据区160个字节，就完成了向PPU的$2040～$21BF共384个单元置入数据的任务，绘出了如图5－5的游戏标题画面。有关画面数据的填写留给读者练习。

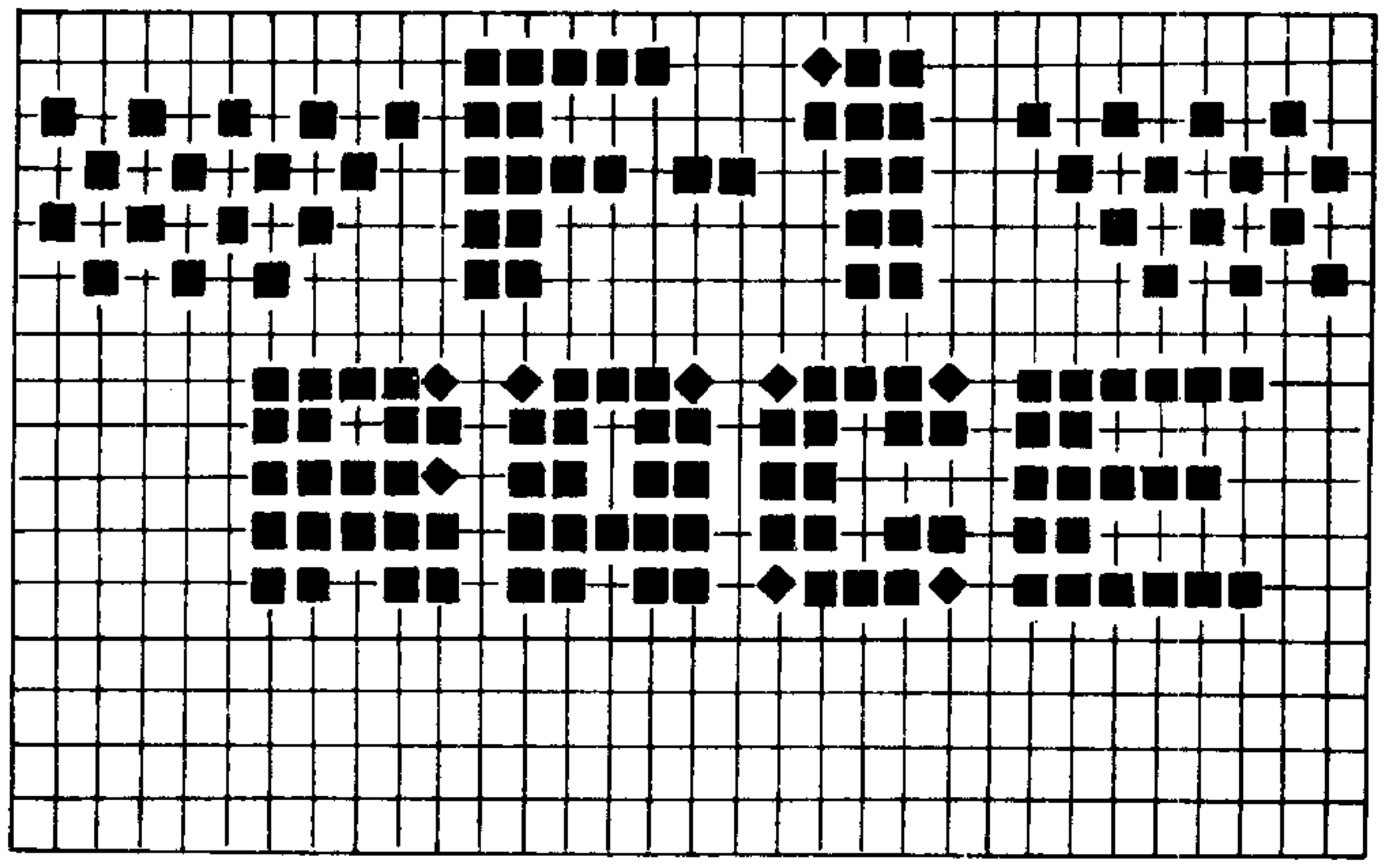


图5－5 《大赛车》标题画面

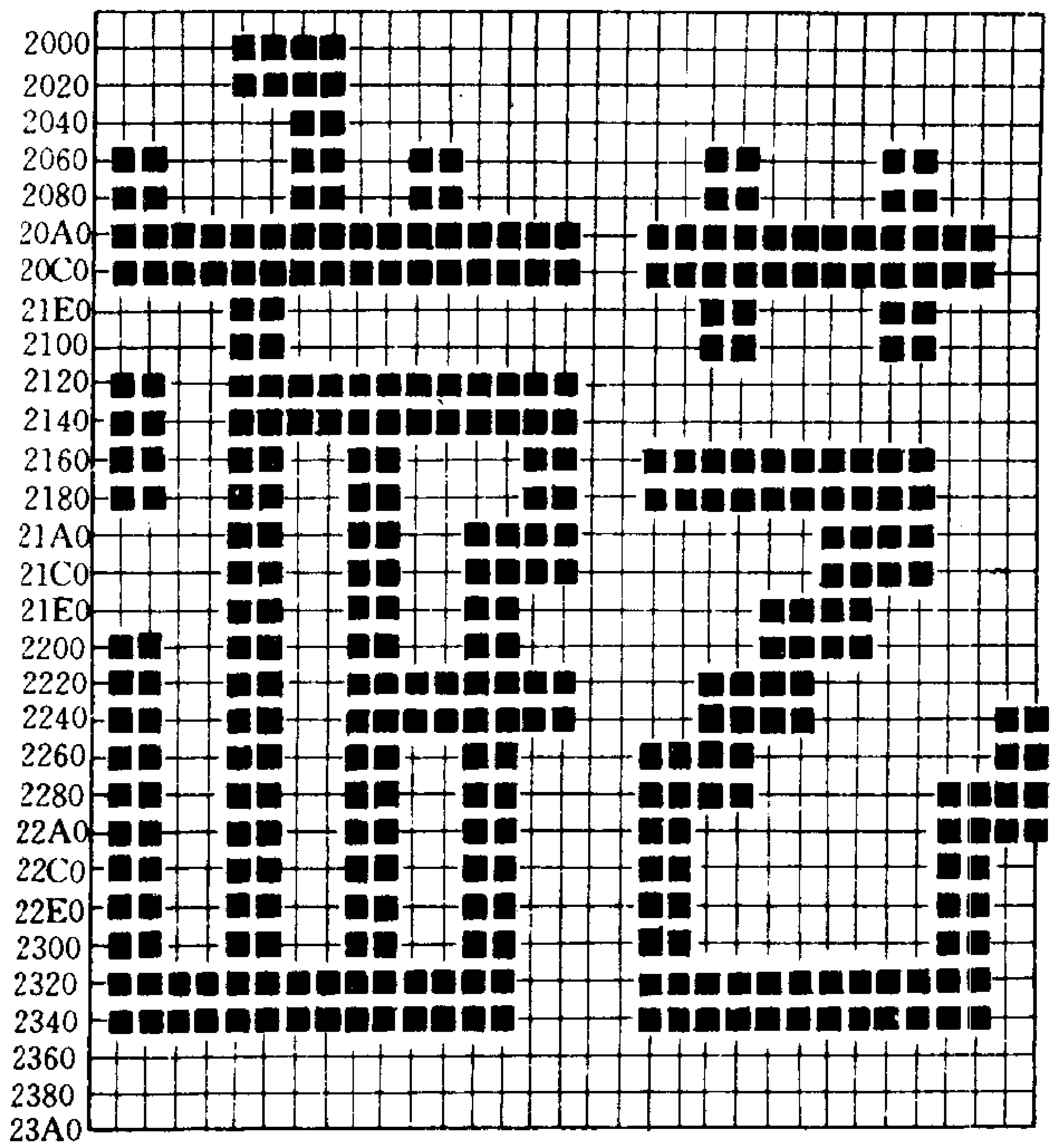


图5－6(A) 背景00页

实际上，如图5－5中的大面积简单画面的绘制还有更简便的方法，称为置色绘图法。《大赛车》游戏的比赛画面中蓝天、绿地以及提示栏的黑色矩形就是采用置色绘图法实现的。绘图方法是：先对整幅画面的显示单元全部置入空格，然后根据绘画要求，将对应显示单元的配色单元置入令该部分画面显示或消失的配色数据，以达到绘图的目的。

下面介绍置色绘图法的程序设计步骤。

我们知道，背景画面中每相连的十六个显示元素使用一个配色单元，那么，使用置色法绘图可使绘图数据区至少压缩十六倍。因此，在大面积简单图形的绘制中，这是一种最优化的绘图方法。下面通过例六详细介绍这种绘图方法的应用。

例六、在背景00页、10页上显示如图5－6的四个巨型汉字“游艺之窗”，并令两页画面循环移位显示。

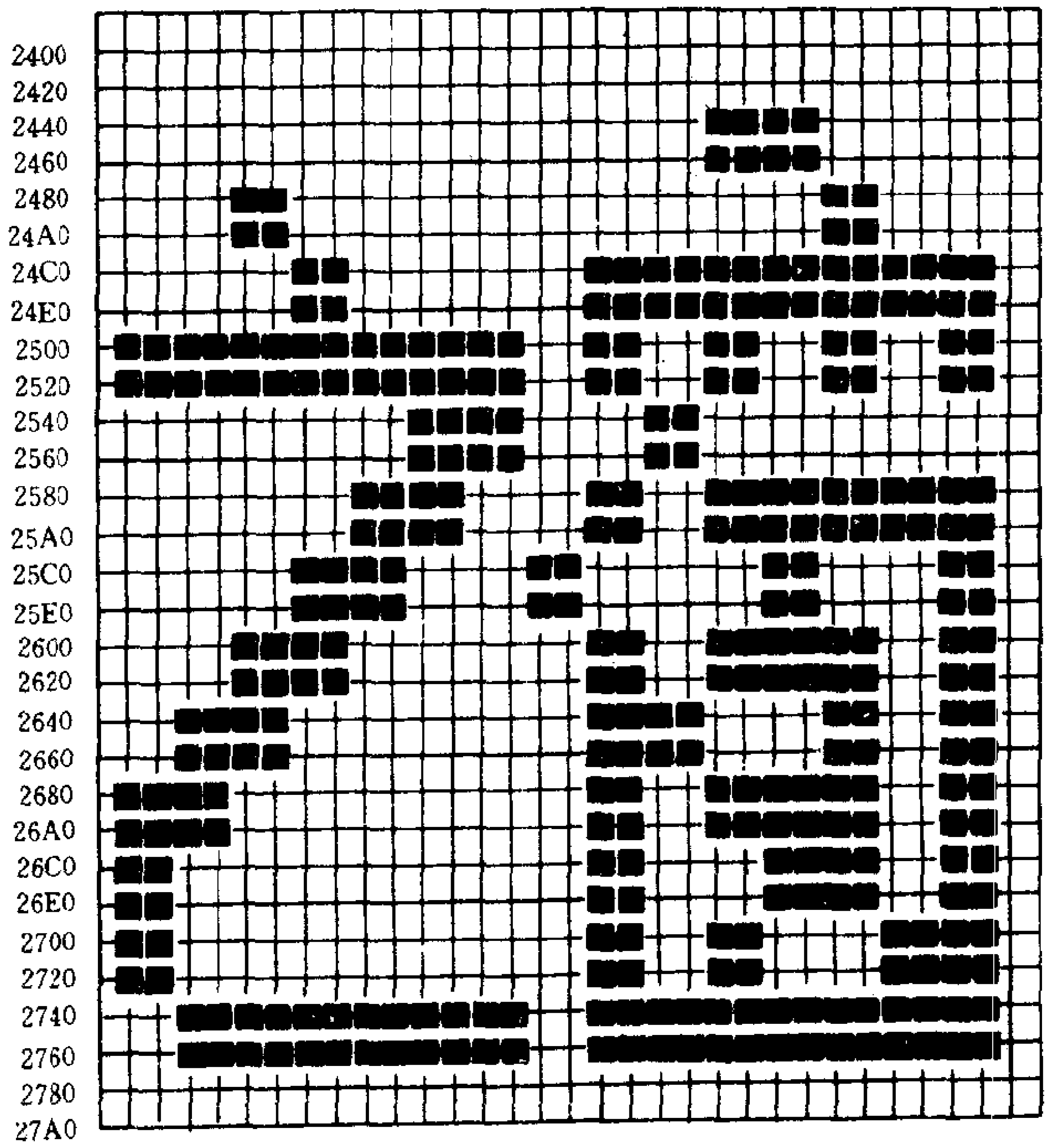


图5－6(B) 背景10页

编程思路：首先定义一个配色代码为全黑色，如选定0＃配色；然后把两背景页的显示映射区全部填满选定的图形，如选定代码为255的光标块；最后把欲显示的图形的配色单元置入黑色，其它置为可见色，也可以反过来把图形部分置为可见色，非图形部分置为黑色。置数操作由机器语言程序完成，画面位移用 F BASIC 程序控制。

编程步骤：

① 把“游艺之窗”四字分别复制到00、10两页的屏幕映射单元的座标图中，见图5－7、图5－8。

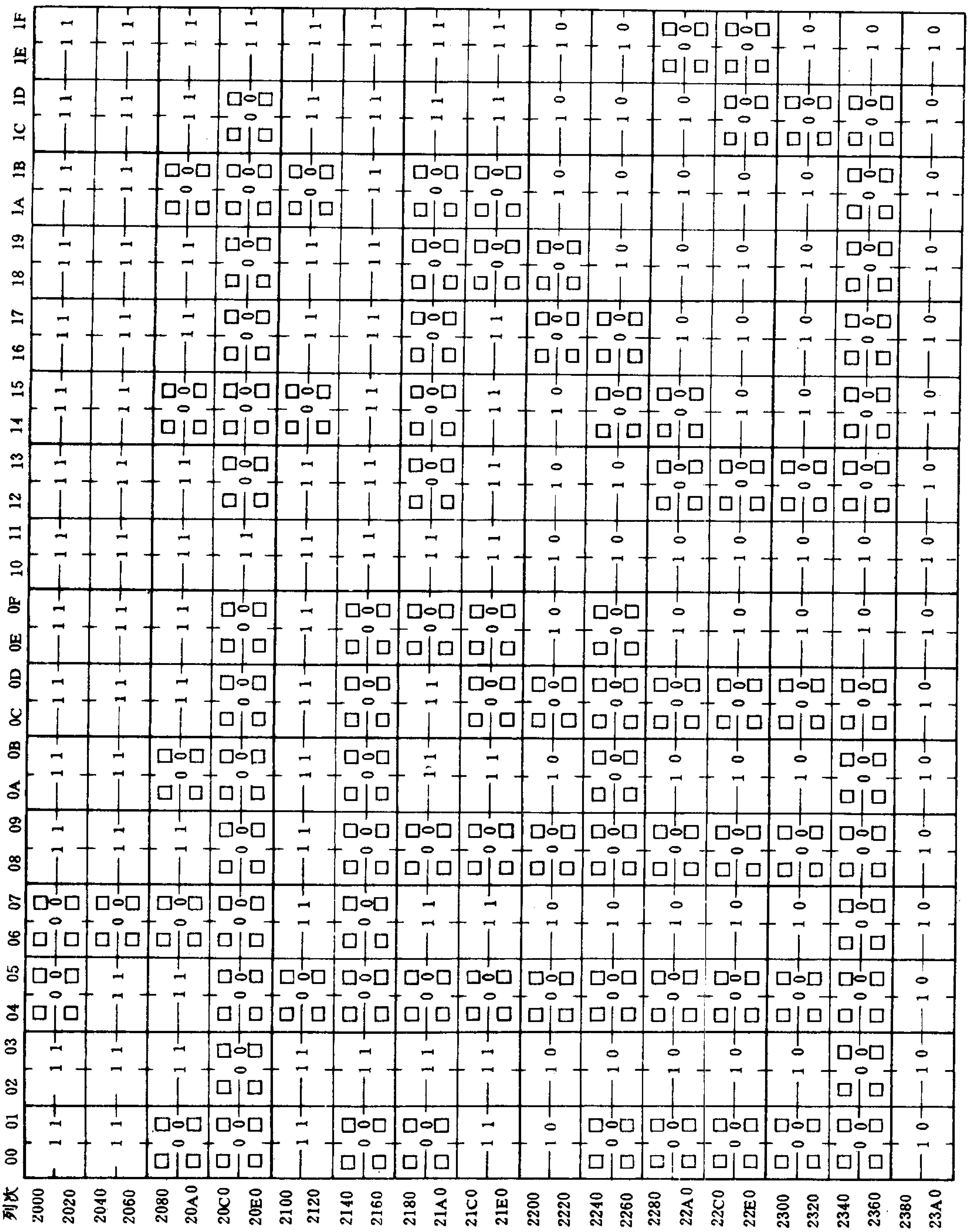


图5－7　背景80页中“游艺”两字的配色座标

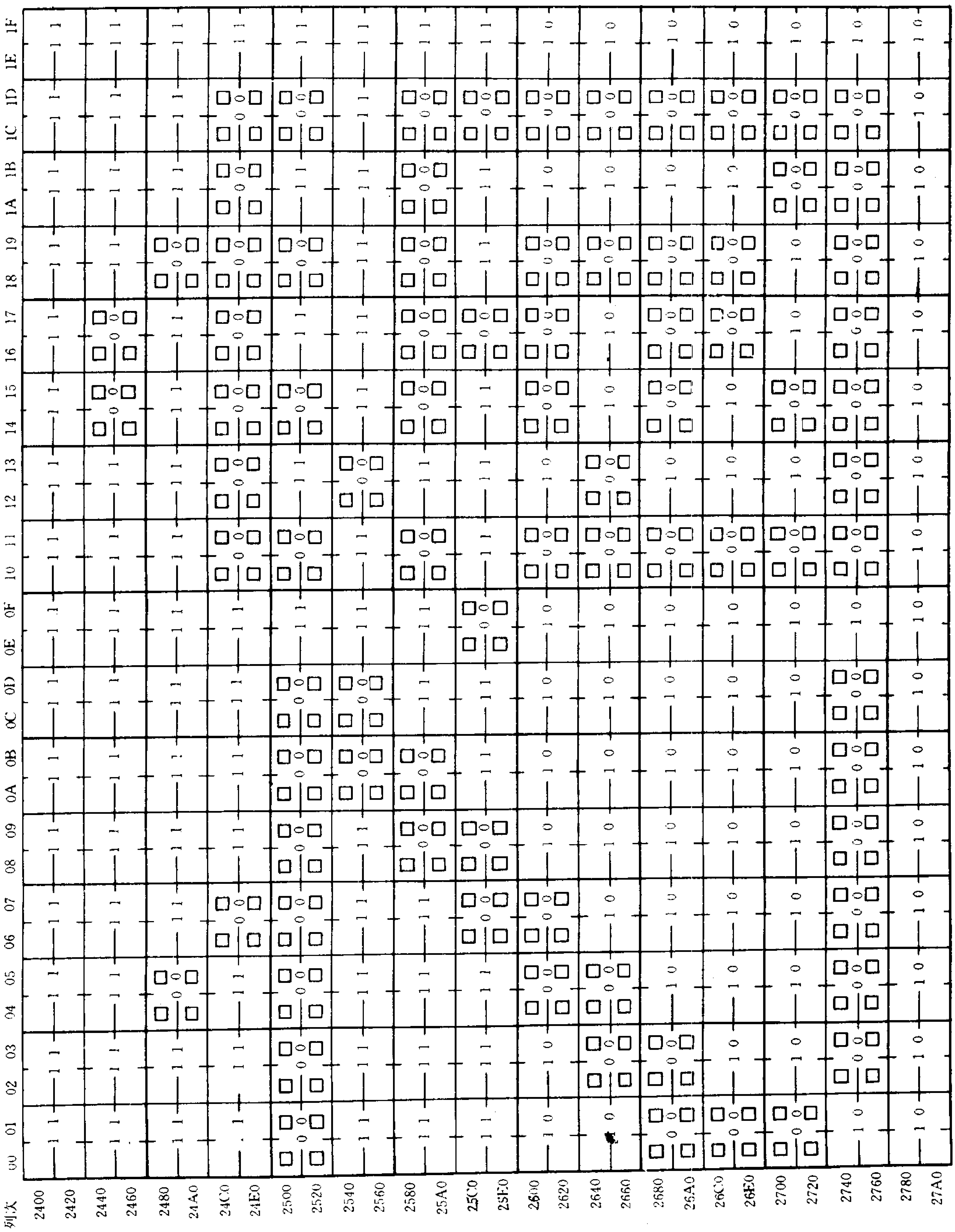


图5—8　背景10页中“之窗”两字的配色座标

② 将图形部分全部填入"00"号配色代码，空白部分填入"11"号配色代码(为使画面更华美，本例的空白区填入了"10、11"两种配色代码。)。

③ 根据各配色单元的配色代码，组合出两显示页面的配色数据。

背景00页的配色数据如下：

```
23C0  FF 30 FF FF FF FF FF FF 0C 03 03 0F 3F 0C 03 CF
23D0  CF 0C 0F 0F FF FC FC FF FC CC CC 03 F3 F0 00 FF
23E0  8A 88 08 08 AA 02 A8 AA 88 88 88 88 22 A8 AA 8A
23F0  08 28 28 A8 22 0A 0A 88 0A 0A 0A 0A 0A 0A 0A 0A
```

背景10页的配色数据如下：

```
27C0  FF FF FF FF FF 0F FF FF FF 3C FF FF 0F 0F 0C CF
27D0  F0 F0 30 CC 3C FC FC FC FF 3F C0 3F FC 30 F0 CC
27E0  2A 80 AA AA 08 A0 88 88 80 AA AA AA 88 20 88 88
27F0  28 0A 0A 8A 08 08 02 88 0A 0A 0A 0A 0A 0A 0A 0A
```

④ 确定 PPU 的配色。我们选取0＃配色代码为黑色、1＃配色代码为蓝色、2＃配色代码为绿色、3＃配色代码为红色，则 PPU 的配色区的颜色代码如下：

```
3F00 0F 0F 0F 0F 0F 02 02 02 0F 0A 0A 0A 0F 05 05 05
```

⑤ 确定程序区、数据区地址。由于程序量和数据量都不太大，故我们选定＄7E00为程序区首址、＄7F00为数据区首址。把 PPU 配色数据写入＄7F00～＄7F0F、背景00页的配色数据写入＄7F10～＄7F4F、背景10页的配色数据写入＄7F50～＄7F8F。

⑥ 编写程序。为便于理解，采取顺序结构编写，读者可把它改写成其它结构形式。机器语言程序见 No. 5－23。

No. 5－23(置色绘图法)背景绘图子程序(三)

```
7E00  48        PHA
7E01  8A        TXA
7E02  48        PHA
7E03  98        TYA
7E04  48        PHA
7E05  A0 00     LDY #$00
7E07  8C 01 20  STY $2001
7E0A  A2 20     LDX #$20
7E0C  8E 06 20  STX $2006
7E0F  8C 06 20  STY $2006
7E12  A0 FF     LDY #$FF
7E14  A9 FF     LDA #$FF
7E16  8D 07 20  STA $2007
7E19  88        DEY
7E1A  D0 FA     BNE $7E16
7E1C  E8        INX
7E1D  E0 28     CPX #$28
7E1F  D0 EB     BNE $7E0C ;将两背景页全部置入 HFF
7E21  A9 3F     LDA #$3F
7E23  A0 00     LDY #$00
7E25  8D 06 20  STA $2006
```

```
7E28  8C 06 20   STY $2006
7E2B  B9 00 7F   LDA $7F00,Y
7E2E  8D 07 20   STA $2007
7E31  C8         INY
7E32  C0 10      CPY #$10
7E34  D0 F5      BNE $7E2B  ;置 PPU 配色
7E36  A9 23      LDA #$23
7E38  8D 06 20   STA $2006
7E3B  A9 C0      LDA #$C0
7E3D  8D 06 20   STA $2006
7E40  B9 00 7F   LDA $7F00,Y
7E43  8D 07 20   STA $2007
7E46  C8         INY
7E47  C0 50      CPY #$50
7E49  D0 F5      BNE $7E40  ;送00页配色
7E4B  A9 27      LDA #$27
7E4D  8D 06 20   STA $2006
7E50  A9 C0      LDA #$C0
7E52  8D 06 20   STA $2006
7E55  B9 00 7F   LDA $7F00,Y
7E58  8D 07 20   STA $2007
7E5B  C8         INY
7E5C  C0 90      CPY #$90
7E5E  D0 F5      BNE $7E55  ;送10页配色
7E60  A5 33      LDA $33
7E62  8D 01 20   STA $2001
7E65  68         PLA
7E66  A8         TAY
7E67  68         PLA
7E68  AA         TAX
7E69  68         PLA
7E6A  60         RTS
```

⑦ 编写 No. 5－23的写入程序和画面位移控制程序。见 No. 5－24。

```
5 REM No. 5-24置色绘图子程序
10 FOR I=0 TO 106 : READ A$ : POKE &7E00+I , VAL("&"+A$) : NEXT
20 FOR I=0 TO 143 : READ A$ : POKE &7F00+I , VAL("&"+A$) : NEXT
30 DATA 48, 8A, 48, 98, 48, A0, 00, 8C, 01, 20, A2, 20, 8E, 06, 20, 8C, 06, 20, A0, FF, A9, FF,
   8D, 07, 20, 88, D0, FA, E8, E0, 28, D0, EB, A9, 3F, A0, 00, 8D, 06, 20, 8C, 06, 20, B9, 00, 7F,
   8D, 07, 20, C8, C0, 10, D0, F5
40 DATA A9, 23, 8D, 06, 20, A9, C0, 8D, 06, 20, B9, 00, 7F, 8D, 07, 20, C8, C0, 50, D0, F5, A9,
   27, 8D, 06, 20, A9, C0, 8D, 06, 20, B9, 00, 7F, 8D, 07, 20, C8, C0, 90, D0, F5, A5, 33, 8D, 01,
   20, 68, A8, 68, AA, 68, 60
50 DATA 0F, 0F, 0F, 0F, 0F, 02, 02, 02, 0F, 0A, 0A, 0A, 0F, 05, 05, 05, FF, 30, FF, FF, FF, FF,
   FF, FF, 0C, 03, 03, 0F, 3F, 0C, 03, CF, CF, 0C, 0F, 0F, FF, FC, FC, FF, FC, CC, CC, 03, F3,
```

```
   F0, 00,
60 DATA FF, 8A, 88, 08, 08, AA, 02, A8, AA, 88, 88, 88, 88, 22, A8, AA, 8A, 08, 28, 28, A8, 22,
   0A, 0A, 88, 0A, 0A, 0A, 0A, 0A, 0A, 0A, 0A, FF, FF, FF, FF, FF, 0F, FF, FF, FF, 3C, FF,
   FF, 0F, 0F, 0C, CF, F0, F0, 30, CC, 3C, FC, FC, FC
70 DATA FF, 3F, C0, 3F, FC, 30, F0, CC, 2A, 80, AA, AA, 08, A0, 88, 88, 80, AA, AA, AA, 88,
   20, 88, 88, 28, 0A, 0A, 8A, 08, 08, 02, 88, 0A, 0A, 0A, 0A, 0A, 0A, 0A, 0A
80 CALL &7E00
90 A=0
100 POKE 8197, A : POKE 8197, 0
110 A=A+1 : IF A>255 THEN 140
120 PAUSE 5
130 GOTO 100
140 IF B=0 THEN POKE 8192, 144 : B=1 : GOTO 90
150 POKE 8192, 145 : B=0 : GOTO 90
160 END
```

输入程序 No. 5－24，检查无误后，运行程序，将看到“游艺之窗”四个巨型大字缓缓左移，周而复始，十分迷人。

程序 No. 5－24仅仅使用了144个数据就完成了两幅画面的绘制，其绘图效率是极高的。若再调用系统子程序并使用双重循环，程序量还可进一步压缩。可见，在大面积简单图形的绘制中，置色绘图法是最优化的一种。

5·4·4 优化查表法

《魂斗罗》游戏已为广大朋友所熟悉、所喜爱。该游戏的编程中有许多优化的方法可供我们学习和借鉴。兹特列两例供参考。

① 武器选项画面的绘制

压缩版的《魂斗罗》游戏开始有一个武器选项画面（见图5－9）。在该画面，游戏者可通过选择键（SELECT）在四种武器中任选其一。该画面的绘制程序见 No. 5－25。

```
            GAME   SELECT
☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

        A.    CONTRA    1
        B.    CONTRA    2
        C.    CONTRA    3
        D.    CONTRA    4
```

图5－9 《魂斗罗》武器选项画面

No. 5－25《魂斗罗》武器选项画面的绘制程序

```
F80D  A94E      LDA  # $4E
F80F  8500      STA  $00
F811  A9F8      LDA  # $F8
F813  8501      STA  $01          ;置绘图数据区首址于$00、$01单元
```

```
F815  A000     LDY # $00
F817  B100     LDA ($00),Y
F819  8D0620   STA $2006       ;置 PPU 地址高位
F81C  C8       INY
F81D  B100     LDA ($00),Y
F81F  8D0620   STA $2006       ;置 PPU 地址低位
F822  C8       INY
F823  B100     LDA ($00),Y
F825  AA       TAX
F826  3011     BMI $F839       ;判断送数指南,若大于 H7F 则
F828  C8       INY             ;转 $F839
F829  B100     LDA ($00),Y     ;送数指南小于 H80的送数处理,
F82B  8D0720   STA $2007       将其后的 N 个数送入 PPU 。
F82E  CA       DEX
F82F  D0F7     BNE $F828
F831  C8       INY
F832  B100     LDA ($00),Y
F834  C980     CMP # $80
F836  D0DF     BNE $F817       ;不是 H80则改变 PPU 地址,继续送数
F838  60       RTS             ;如是 H80则结束送数
F839  8A       TXA
F83A  297F     AND # $7F
F83C  AA       TAX             ;以该数的 D6~D0位为计数器,将其后
F83D  C8       INY             的一数连送 N 次。
F83E  B100     LDA ($00),Y
F840  8D0720   STA $2007
F843  CA       DEX
F844  D0FA     BNE $F840
F846  F0E9     BEQ $F831
```

数据区如下:

```
F84E                                            3F 00
                                                PPU 配
F850  08 0F 20 20 20 0F 27 27 27 3F 10 04 0F 05 0F 17
      色, 送 $3F00~$3F07。     送 PPU  $3F10~$3F13
F860  23 D0 88 50 20 E4 17 20 20 20 20 20 20 20 20 20
      背景零页配色, $23D0~$23D7
F870  20 20 20 20 20 20 20 20 20 20 20 20 20 20 21 2A
F880  0B 47 41 4D 45 00 53 45 4C 45 43 54 21 66 93 3F
         G  A  M  E     S  E  L  E  C  T
F890  21 AA 0B 41 40 00 43 4F 4E 54 52 41 00 31 21 EA
               A  .     C  O  N  T  R  A     1
F8A0  0B 42 40 00 43 4F 4E 54 52 41 00 32 22 2A 0B 43
         B  .     C  O  N  T  R  A     2              C
```

```
F8B0  40 00 43 4F 4E 54 52 41 00 33 22 6A 0B 44 40 00
        .     C  O  N  T  R  A     3           D  .
F8C0  43 4F 4E 54 52 41 00 34 00 00
         C  O  N  T  R  A     4
```

图5—9的数据区仅使用了124个单元就完成了PPU配色、背景页配色和画面显示，虽然四行都是相同的内容(CONTRA)，但数据区中已留出了相应的空间，即使送显其它内容，数据区也不会增加。

程序No.5—25的结构比较简单，读者可自己完成程序的转换，不多赘述。

②《魂斗罗》标题画面的绘制

我们知道，《魂斗罗》的标题画面是由画写的"CONTRA"和两个主人翁的上身组成的。画面既简单抽象又富有力的感召，是正义和美的有机结合与升华。《魂斗罗》仅凭一幅标题画面就勾起了人们的无限遐思与向往，这怎能不令人钦佩和惊叹呢！现在让我们看看这一标题画面是如何绘制的。

标题画面的绘制程序在＄C9CF～＄CA60，其数据区在第二体的＄9097～＄9251，现抄录如下，见程序No.5—26。

No.5—26《魂斗罗》标题画面绘制程序

```
C9CF  A001     LDY #$01       ;该子程序调用前已向$00、$01单元
C9D1  B100     LDA ($00),Y    置入数据区首址$9097
C9D3  8D0620   STA $2006      ;置PPU地址高位
C9D6  88       DEY
C9D7  B100     LDA ($00),Y
C9D9  8D0620   STA $2006      ;置PPU地址低位
C9DC  A902     LDA #$02
C9DE  A604     LDX $04
C9E0  1001     BPL $C9E3
C9E2  0A       ASL            ;若$04>H7F则A=4
C9E3  A200     LDX #$00
C9E5  2092C8   JSR $C892      ;调整取数地址。
C9E8  A000     LDY #$00
C9EA  B100     LDA ($00),Y
C9EC  C9FF     CMP #$FF       ;若A=HFF则结束送数
C9EE  F06B     BEQ $CA5B
C9F0  C97F     CMP #$7F       ;若A=H7F则改变PPU地址
C9F2  F05D     BEQ $CA51
C9F4  A8       TAY
C9F5  1026     BPL $CA1D
C9F7  297F     AND #$7F
C9F9  8502     STA $02        ;若取数指南小于H7F则以$02为计数器，
C9FB  A001     LDY #$01       送后面的N个数。
C9FD  B100     LDA ($00),Y
C9FF  A604     LDX $04
CA01  1003     BPL $CA06
CA03  2036CA   JSR $CA36      ;若$04>H7F则延时送数
```

```
CA06  8D0720     STA  $2007
CA09  C402       CPY  $02
CA0B  F003       BEQ  $CA10          ;达到送数数量则转$CA10调整地址
CA0D  C8         INY
CA0E  D0ED       BNE  $C9FD          ;未达到送数数量则转$C9FD 继续送数
CA10  A901       LDA  #$01
CA12  18         CLC
CA13  6502       ADC  $02
CA15  A200       LDX  #$00
CA17  2092C8     JSR  $C892          ;调整取数地址
CA1A  4CE8C9     JMP  $C9E8
CA1D  A001       LDY  #$01           ;当取数指南>H7F 时,把后面的一个数据
CA1F  8502       STA  $02            连送 N 次的处理。
CA21  B100       LDA  ($00),Y
CA23  A402       LDY  $02
CA25  A604       LDX  $04
CA27  1003       BPL  $CA2C
CA29  2036CA     JSR  $CA36
CA2C  8D0720     STA  $2007
CA2F  88         DEY
CA30  D0FA       BNE  $CA2C
CA32  A902       LDA  #$02
CA34  D0DF       BNE  $CA15          ;一次送数结束后,调整取数地址
CA36  8503       STA  $03            ;以下是延时子程序
CA38  0603       ASL  $03
CA3A  6A         ROR
CA3B  0603       ASL  $03
CA3D  6A         ROR
CA3E  0603       ASL  $03
CA40  6A         ROR
CA41  0603       ASL  $03
CA43  6A         ROR
CA44  0603       ASL  $03
CA46  6A         ROR
CA47  0603       ASL  $03
CA49  6A         ROR
CA4A  0603       ASL  $03
CA4C  6A         ROR
CA4D  0603       ASL  $03
CA4F  6A         ROR
CA50  60         RTS
CA51  A901       LDA  #$01
CA53  A200       LDX  #$00
CA55  2092C8     JSR  $C892          ;调整取数地址
```

```
CA58  4CCFC9       JMP  $C9CF
CA5B  60           RTS                ;结束送数
```

调整取数地址子程序

```
C892  18           CLC
C893  7500         ADC  $00,X
C895  9500         STA  $00,X
C897  9002         BCC  $C89B
C899  F601         INC  $01,X
C89B  60           RTS
```

绘图数据

```
9097                        00 20 4B 00 8B 00 00 00 00
                            PPU $2000 先送 H4B 个00、又送 H0B 个00
90A0  00 00 00 00 00 00 00 14 00 8C 00 00 00 00 00 00
                            送 H14个00，送 H0C 个00（下同，不再注释）
90B0  00 00 00 00 00 00 14 0082 00 00 3A 00 86 02 03
90C0  04 05 06 07 19 00 89 11 12 13 14 15 16 17 18 19
90D0  16 00 8A 20 21 22 23 24 25 26 27 28 29 16 00 83
90E0  60 61 62 04 00 82 08 09 17 00 96 70 71 72 00 00
90F0  63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 0A 0B 0C
9100  0D 0A 00 97 80 81 82 00 00 73 74 75 76 77 78 79
9110  7A 7B 7C 7D 7E 7F 1A 1B 1C 1D 1E 09 00 98 90 91
9120  92 00 00 83 84 85 86 87 88 89 8A 00 8C 8D 8E 8F
9130  2A 2B 2C 2D 2E 2F 08 00 98 A0 A1 A2 A3 00 93 94
9140  95 96 97 98 99 9A 00 9C 9D 9E 9F 3A 3B 3C 3D 3E
9150  3F 09 00 8B B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB 17
9160  00 86 A4 A5 A6 A7 A8 A9 27 00 85 C4 00 D0 D1 D2
9170  1A 00 87 D3 D4 D5 E0 E1 E2 E3 0B 00 8B 50 4C 41
                                                P  L  A
9180  59 00 53 45 4C 45 43 54 04 00 85 E4 E5 F0 F1 F2
      Y     S  E  L  E  C  T
9190  1A 00 88 F3 F4 F5 C5 C6 C7 C8 C9 0C 00 88 31 00
91A0  50 4C 41 59 45 52 03 00 85 CA CB CC CD CE 03 00
      P  L  A  Y  E  R
91B0  82 EF D6 15 00 86 D9 DA DB DC DD DE 04 00 81 E6
91C0  0B 00 90 32 00 50 4C 41 59 45 52 53 00 E9 EA EB
               2     P  L  A  Y  E  R  S
91D0  EC ED EE 01 00 81 F6 17 00 84 D7 D8 E7 E8 13 00
91E0  8D 00 00 00 00 00 00 00 00 00 00 00 00 00 0E 00
91F0  97 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9200  00 00 00 00 00 00 00 00 10 00 8B 00 00 00 00 00
9210  00 00 00 00 00 00 0E 00 97 00 00 00 00 00 00 00
9220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00（以下为配色数据）
9230  4D 00 83 A0 A0 20 06 00 82 58 52 03 50 83 00 00
9240  20 05 05 05 00 83 CC FF F3 05 00 83 FF FF CC 11
```

列次	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
2000																																
2020																																
2040																																
2060																																
2080																																
20A0																																
20C0							02	03	04	05	06	07																				
20E0						11	12	13	14	15	16	17	18	19																		
2100					20	21	22	23	24	25	26	27	28	29																		
2120					60	61	62					08	09																			
2140					70	71	72			63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	0A	0B	0C	0D						
2160					80	81	82			73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	1A	1B	1C	1D	1E					
2180					90	91	92			83	84	85	86	87	88	89	8A		8C	8D	8E	8F	2A	2B	2C	2D	2E	2F				
21A0					A0	A1	A2	A3		93	94	95	95	97	98	99	9A		9C	9D	9E	9F	3A	3B	3C	3D	3E	3F				
21C0						B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB																
21E0								A4	A5	A6	A7	A8	A9																			
2200																					C4		D0	D1	D2							
2220																				D3	D4	D5	E0	E1	E2	E3						
2240						50	4C	41	59		53	45	4C	45	43	54					E4	E5	F0	F1	F2							
2260																				F3	F4	F5	C5	C6	C7	C8	C9					
2280								31		50	4C	41	59	45	52				CA	CB	CC	CD	CE				EF	D6				
22A0																		D9	DA	DB	DC	DD	DE					E6				
22C0								32		50	4C	41	59	45	52	53		E9	EA	EB	EC	ED	EE					F6				
22E0																				D7	D8	E7	E8									
2300																																
2320																																
2340																																
2360																																
2380																																
23A0																																

注———空格部分为“00”

图5—10 《魂斗罗》标题画面的图形代码座标图

9250 00 FF

注——数据区中，凡下面划双线的数据为取数指南。

程序 No. 5－26使用了442个数据完成了标题画面的绘制。实际上单对标题画面而言，数据量还可以压缩。如在清屏以后绘图时，数据区中的 $ 9097～ $ 90BC 和 $ 91E0～ $ 922F 这118个置零(清屏)的数据可以删除，那么数据区将减少到326个单元。可见，程序 No. 5－26是较优化的绘图程序。

标题画面的图形代码排列见图5－10。

为使朋友们能够直接使用《魂斗罗》的绘图方法设计背景图案，特改造程序 No. 5－26为 No. 5－27。(仍以绘制《魂斗罗》的标题画面为例，使用原数据区。)

No. 5－27背景绘图子程序(四)

```
7F00  206C7F   JSR  $7F6C        ;保护现场，置数据区首址于$50、$51
7F03  A900     LDA  #$00
7F05  8D0120   STA  $2001        ;关显示
7F08  A001     LDY  #$01
7F0A  B150     LDA  ($50),Y
7F0C  8D0620   STA  $2006        ;置 PPU 地址高位
7F0F  88       DEY
7F10  B150     LDA  ($50),Y
7F12  8D0620   STA  $2006        ;置 PPU 地址低位
7F15  A902     LDA  #$02
7F17  20647F   JSR  $7F64        ;调整取数地址
7F1A  A000     LDY  #$00
7F1C  B150     LDA  ($50),Y
7F1E  C9FF     CMP  #$FF
7F20  D004     BNE  $7F26        ;A≠HFF 则转$7F26取数送入 PPU
7F22  20727F   JSR  $7F72        ;A=HFF 则结束送数，恢复现场，开显示
7F25  60       RTS
7F26  C97F     CMP  #$7F
7F28  F032     BEQ  $7F5C        ;A=H7F 转$7F5C 改变 PPU 地址
7F2A  A8       TAY
7F2B  101D     BPL  $7F4A        ;A<H80转$7F4A 进行一个数
7F2D  297F     AND  #$7F         连送 N 次的处理；A>H7F 时则以其
7F2F  8552     STA  $52          D6～D0位为计数送入$52
7F31  A001     LDY  #$01
7F33  B150     LDA  ($50),Y      ;进行取 N 个数送入 PPU 的处理
7F35  8D0720   STA  $2007
7F38  C452     CPY  $52
7F3A  F003     BEQ  $7F3F        ;达到数量则调整取数地址
7F3C  C8       INY
7F3D  D0F4     BNE  $7F33        ;未达到则继续取数送数
7F3F  A901     LDA  #$01
7F41  18       CLC
7F42  6552     ADC  $52
```

```
7F44  20647F   JSR $7F64      ;调整取数地址
7F47  4C1A7F   JMP $7F1A
7F4A  A001     LDY #$01       ;一个数连送N次的处理
7F4C  8552     STA $52
7F4E  B150     LDA ($50),Y
7F50  A452     LDY $52
7F52  8D0720   STA $2007
7F55  88       DEY
7F56  D0FA     BNE $7F52
7F58  A902     LDA #$02
7F5A  D0E8     BNE $7F44      ;送完后调整取数地址
7F5C  A901     LDA #$01
7F5E  20647F   JSR $7F64      ;调整取数地址
7F61  4C087F   JMP $7F08      ;调整PPU地址
7F64  18       CLC            ;调整取数地址子程序
7F65  6550     ADC $50
7F67  9002     BCC $7F6B
7F69  E651     INC $51
7F6B  60       RTS
7F6C  48       PHA            ;保护现场子程序
7F6D  8A       TXA
7F6E  48       PHA
7F6F  98       TYA
7F70  48       PHA
7F71  A997     LDA #$97
7F73  8550     STA $50
7F75  A990     LDA #$90
7F77  8551     STA $51
7F79  60       RTS
7F7A  A531     LDA $31        ;恢复现场子程序
7F7C  8D0120   STA $2001
7F7F  68       PLA
7F80  A8       TAY
7F81  68       PLA
7F82  AA       TAX
7F83  68       PLA
7F84  60       RTS
```

该子程序不必放入中断即可运行,读者只要把原数据区放入RAM的适当位置,运行No.5－27就可以绘出《魂斗罗》标题画面的图形结构(由于字模不同,绘出的画面是不一样的)。

5·4·5　间接分段绘图法(《魂斗罗》游戏画面的基本绘制方法)

《魂斗罗》中的游戏画面都是采取间接分段绘图法绘制的。这种绘图方法的特点是:整幅画面分散绘制,一般一次中断或隔几次中断绘制一列或一行,横向卷动的画面每次绘一列,纵向

滚动的画面每次绘一行；以七页为间接绘图数据缓存区，由当前中断把绘图数据送入七页，在下一次中断再把存于七页的数据送入 PPU。这种绘图方法可保证多幅画面连续绘制，产生连绵不绝的效果。如第一关开始时的游戏画面尤如幕布从左向右拉开，而第二关又如幕布由下向上提起。随着游戏的进程，画面一幕幕的位移，使人丝毫感觉不到画面的间断，整关游戏画面好象是一幅长卷图画，左卷右放，尽头方休。下面仅列出源绘图程序并加以简要说明，以供学习、参考。源程序见 No. 5－28。

程序 No. 5－28共分两段：第一段为＄CB60～＄CBC5，完成把 PPU 地址、绘图数据从数据区读出，按照一定的格式送入七页；第二段为＄CC3F～＄CC7E，完成把七页的绘图数据送入 PPU，绘制游戏画面。

No. 5－28《魂斗罗》游戏画面绘制源程序

从数据区取数送入七页子程序

```
CB60  48        PHA              ;保存调该程序时的 A 值，
CB61  A902      LDA #$02         该值决定绘图的种类
CB63  8503      STA $03
CB65  A901      LDA #$01
CB67  20BDCB    JSR $CBBD        ;令$700=1、$21=1
CB6A  68        PLA              ;弹出绘图类型数据
CB6B  8502      STA $02
CB6D  0A        ASL
CB6E  AA        TAX              ;以绘图类型数据为地址偏移量
CB6F  BD62B2    LDA $B262,X
CB72  8500      STA $00
CB74  BD63B2    LDA $B263,X
CB77  8501      STA $01          ;读出绘图数据区首址存入
CB79  A621      LDX $21          $00、$01单元
CB7B  A000      LDY #$00
CB7D  B100      LDA ($00),Y      ;读第一个数据
CB7F  C8        INY
CB80  C9FF      CMP #$FF
CB82  F03F      BEQ $CBC3        ;若 A=HFF 则结束读数
CB84  C9FE      CMP #$FE
CB86  F019      BEQ $CBA1        ;若 A=HFE 则令$700+X=HFF、
CB88  C9FD      CMP #$FD         X+1→$21,结束取数
CB8A  F019      BEQ $CBA5        ;若 A=HFD 则令$700+X=HFF,
CB8C  9D0007    STA $0700,X      ;进行第二次送数，把数据送入七页
CB8F  A502      LDA $02
CB91  100B      BPL $CB9E
CB93  A503      LDA $03          ;七页的数据格式为：
CB95  D005      BNE $CB9C        $700表示送数方式；
CB97  9D0007    STA $0700,X      $701表示一组送入 PPU 的数据个数
CB9A  F002      BEQ $CB9E        $702表示送入 PPU 数据的组数
CB9C  C603      DEC $03          $703表示 PPU 地址高位
CB9E  E8        INX              $704表示 PPU 地址低位
```

```
CB9F  D0DC      BNE $CB7D        $705……为绘图数据
CBA1  A9FF      LDA #$FF         当一次送多组数据时,每一组数据
CBA3  D01A      BNE $CBBF        传送完毕后的下两个数据,即为下一组
CBA5  A9FF      LDA #$FF         绘图数据的 PPU 地址。
CBA7  20BFCB    JSR $CBBF
CBAA  A902      LDA #$02
CBAC  8503      STA $03
CBAE  A901      LDA #$01
CBB0  20BFCB    JSR $CBBF
CBB3  D0C8      BNE $CB7D
CBB5  A9FF      LDA #$FF
CBB7  D004      BNE $CBBD
CBB9  A900      LDA #$00
CBBB  F000      BEQ $CBBD
CBBD  A621      LDX $21
CBBF  9D0007    STA $0700,X
CBC2  E8        INX
CBC3  8621      STX $21
CBC5  60        RTS
```

取七页数据送入 PPU 子程序

```
CC3F  A200      LDX #$00
CC41  A5FF      LDA $FF
CC43  2918      AND #$18
CC45  8502      STA $02
CC47  BC0007    LDY $0700,X
CC4A  F0D0      BEQ $CC1C        ;$700=0不送数
CC4C  A502      LDA $02
CC4E  19C5CB    ORA $CBC5,Y
CC51  8D0020    STA $2000        ;设置送数方式
CC54  E8        INX
CC55  BD0007    LDA $0700,X
CC58  8500      STA $00          ;以$00为一组送数的个数计数器
CC5A  E8        INX
CC5B  BD0007    LDA $0700,X
CC5E  8501      STA $01          ;以$01为本次送数的组数计数器
CC60  A400      LDY $00
CC62  E8        INX
CC63  BD0007    LDA $0700,X
CC66  8D0620    STA $2006        ;置 PPU 地址高位
CC69  E8        INX
CC6A  BD0007    LDA $0700,X
CC6D  8D0620    STA $2006        ;置 PPU 地址低位
CC70  E8        INX
CC71  BD0007    LDA $0700,X
```

```
CC74  8D0720    STA $2007       ;取数送入 PPU
CC77  88        DEY
CC78  D0F6      BNE $CC70       ;若本组未送完则转$CC70继续
CC7A  C601      DEC $01         ;若本组已送完则令组数计数器减一
CC7C  D0E2      BNE $CC60       ;若还有数据组则转$CC60继续送数
CC7E  60        RTS             ;若数据组计数器为零则结束送数
```

程序 No.5－28通俗易懂，留给读者练习编写用于电脑游戏机的实用程序。

本节从任天堂游戏编程中最常用的背景绘图方法入手，详细分析讨论了十八个背景绘图程序，并给出了可直接用于电脑游戏机的背景绘图实用子程序，使即使不懂机器语言的朋友也能随心所欲的设计自己喜爱的画面。当然，任天堂游戏的实际背景编程都要比这些子程序复杂的多，本节所列举的子程序都是进行简化的。可能有些朋友觉得不够，想了解的更多一些。但是，由于电脑游戏机的用户内存有限，和其工作系统的能力所限，我们不可能编写较大的游戏程序，而且本书的中心目的重点在于教授电脑游戏机背景绘图的基本方法，旨在抛砖引玉，引导喜爱游戏电脑的朋友们设计出效率更高的绘图程序。至于更高层次的任天堂游戏背景处理的方法技巧，如画面分割、局部位移、多屏卷动等控制技术，已超出了本书的讨论范围，这些我们将在《任天堂游戏编程探密》一书中详细讨论。（目前，北京裕兴机电研究所推出了带有掉电保持功能的、有32K RAM 的超级学习卡，若使用得当，将有助于我们编写较大的游戏程序。）

5·5　卡通控制编程技巧

任天堂游戏中的角色被统称为卡通。玩任天堂游戏的过程就是控制卡通动作的过程。众所周知，卡通在整个游戏的过程中是最为活跃的部分，因而控制卡通的程序也就最为复杂。一般情况下，游戏软件中用于控制卡通动作的程序量约占总程序量的百分之五十左右（不包括数据区），游戏越精彩，所占比例越大。如大家最宠爱的枪战游戏《魂斗罗》，软件容量为128K，其中控制程序约占32K，而用于卡通控制的程序则占到20K 之多，比例高达百分之六十以上。由此可见，卡通动作的控制是十分复杂的。因此，限于篇幅，我们不可能对卡通的动作控制程序作全面的介绍。只能教给大家最基本的方法，由朋友们在编程实践中举一反三，领会掌握。本节将首先指出 F BASIC 系统中卡通与任天堂游戏中卡通的区别；然后重点讨论操纵器的采样程序设计和卡通的定义方法；最后列出《大赛车》游戏中定义卡通的部分程序清单供欣赏参考。

5·5·1　两个系统中卡通的区别

我们已经熟悉了 F BASIC 工作系统中的卡通，并能得心应手的编写程序控制卡通的运动和显示，以实现不同的功能要求。同时，我们还都有深切的体会：用 F BASIC 语言编写的卡通控制程序，运行起来速度特别慢，比任天堂游戏中的卡通要逊色的多，而且不能定义高点阵的卡通，无论怎样绞尽脑汁修改程序也难尽人意。因而，迫切地想了解任天堂游戏中对卡通的控制方法，梦寐以求地想把这种控制方法应用到 F BASIC 的工作系统中来。但是，任天堂游戏中的控制方法又独有特点，只有全面掌握两个系统对卡通控制方法的相似与区别，才能够应用灵活，行之有效。现将 F BASIC 、6527CPU（任天堂游戏机）两个工作系统中卡通的特点归纳如下：

1、在 F BASIC 中，卡通被分为显示、动作两类，分别用 DEF SPRITE、DEF MOVE 指令来

定义这两类卡通的显示和运动。而在任天堂游戏中,卡通是没有显示、动作之分的,它里面的卡通具有相同的共性和个性。F BASIC 把它分为两类的目的是为了便于指令控制,以实现不同的功能要求。这应该说是 F BASIC 对游戏机功能的扩展和创新。

2、在任天堂游戏中,一般使用二页作为卡通的定义区。由于每一个卡通块(卡通显示的最小单位)要占用四个单元,故一次最多可定义64个卡通块。F BASIC 规定16点阵显示、动作卡通的数量各不允许超过八个;在任天堂游戏中则没有这种限制。只要一次定义的卡通块数量不超过64个,就可全部显示。

3、在 F BASIC 中限定卡通最大为16×16点阵;在任天堂游戏中则没有这种限制。只要实际需要,系统允许把64个卡通块定义成一个特大卡通。

4、在 F BASIC 中,卡通一次定义后可一直使用,只要不改变它的运动状态,就不必重新定义;而在任天堂游戏中,卡通必须反复定义才能不断运动。即在 F BASIC 中,卡通的运动是由系统控制而自动运动的;而在任天堂游戏中,卡通的每一点运动都必须靠编程控制。因此,单从这一方面来说 F BASIC 中的卡通属智能型的;任天堂中的卡通属机械型的。但是,智能型的卡通由于程序已经固化,运动形式已经锁定,不能人工干预改变,因而它的原有优势反而变成了束缚它的桎梏,使它变成了机械型的。而任天堂中机械型的卡通,由于可通过编程控制其运动状态,因而,可纵横捭阖,灵活控制。从而扭转了它原来的机械劣势,而体现了领先的优势。

5、在 F BASIC 中,运动卡通的最快速度为每次一点,即每中断一次运动一点,运动一点的最短时间为0.02秒。另外,卡通的运动方向只有系统设定好的八个(0°、45°、90°、135°、180°、225°、270°、315°),用户不能再作其它选择。而任天堂中,每中断一次可以对卡通重新定义一次,因而只要适当改变卡通的定义座标,就可以任意控制卡通的运动速度和运动方向。

6、F BASIC 中16×16点阵卡通的组合是系统约定好的,用户不能改变。如我们只能定义面向左或面向右、头朝上或头朝下的马里欧,而不能定义睡倒在地脸朝上或朝下的马里欧。同样也不能定义摔倒的丽莎或企鹅。虽然我们可以使用 DEF SPRITE 语句来选择组合,但组合的卡通又不能自动运动,因此其使用范围受到了极大的限制。造成这些束缚的原因是:卡通的字模数据已经设定,不能随意改变,用户只能使用系统固有的256个卡通字模(卡通块)组合卡通。因此,F BASIC 中的卡通是呆板的。在任天堂游戏中,卡通字模可随时改变,如《魂斗罗》几乎每一关的开始、关底都要更换字模,其字模数据区约占45K。而《魂斗罗》二代、《脱狱》二代、《双截龙》等256K 的游戏,其字模区竟高达128K。就因为这些游戏有如此丰富的字模,所以才能产生那么奇妙的背景画面和形神并茂的奇特卡通。

5·5·2 操纵器的采样

在任天堂游戏的过程中,操纵器就是司令部,游戏者的意志就是通过操纵器传送给游戏机的。游戏者游戏水平的高低也全部从控制操纵器的过程中体现出来。可见,操纵器的采样处理直接关系到游戏者意志的完整转达。因此,操纵器采样处理程序性能的好坏直接体现了一个游戏软件的设计水平,也直接影响着游戏软件的功能评价。

操纵器的采样是通过读系统软开关(寄存器)$4016、$4017实现的。针对游戏控制过程的特点,一般要求能够读出操纵器的瞬间值,以满足快速变化控制的需要;能够读出操纵器的稳键值,以满足选项过程中控制选项的需要;同时还要求能够输出操纵器的单步值和连续值,以实现不同的游戏控制功能。

每个软件的操纵器采样程序模式虽然比较相近,但也各有千秋。下面仅介绍《大赛车》、《小

学算术》、《魂斗罗》游戏中三个较典型的操纵器采样程序，最后给出 BS. 2A F BASIC 系统操纵器采样子程序的调用方法。

1、《大赛车》的操纵器采样

《大赛车》为单人游戏，游戏中只对0＃操纵器采样。程序如 No. 5－29。

No. 5－29《大赛车》操纵器采样程序

```
C066  48      PHA
C067  8A      TXA
C068  48      PHA
C069  98      TYA
C06A  48      PHA
C06B  A900    LDA # $00
C06D  8D0320  STA $2003
C070  A54F    LDA $4F
C072  2901    AND # $01
C074  0902    ORA # $02
C076  8D1440  STA $4014
C079  A001    LDY # $01
C07B  8C1640  STY $4016
C07E  88      DEY
C07F  8C1640  STY $4016
C082  AD1640  LDA $4016
C085  2903    AND # $03
C087  856D    STA $6D        ;A 键代码存 $6D
C089  AD1640  LDA $4016
C08C  2903    AND # $03
C08E  856E    STA $6E        ;B 键代码存 $6E
C090  AD1640  LDA $4016
C093  2903    AND # $03
C095  856B    STA $6B        ;SELECT 键代码存 $6B
C097  AD1640  LDA $4016
C09A  2903    AND # $03
C09C  856C    STA $6C        ;START 键代码存 $6C
C09E  A001    LDY # $01
C0A0  AD1640  LDA $4016
C0A3  2903    AND # $03
C0A5  D003    BNE $C0AA
C0A7  88      DEY
C0A8  10F6    BPL $C0A0
C0AA  846F    STY $6F        ;上、下键代码存 $6F
C0AC  A001    LDY # $01
C0AE  AD1640  LDA $4016
C0B1  2903    AND # $03
C0B3  D003    BNE $C0B8
```

```
C0B5  88      DEY
C0B6  10F6    BPL  $C0AE
C0B8  8470    STY  $70          ;左、右键代码存$70
C0BA  A471    LDY  $71
C0BC  F00A    BEQ  $C0C8
C0BE  846D    STY  $6D
C0C0  88      DEY
C0C1  846E    STY  $6E
C0C3  88      DEY
C0C4  8470    STY  $70
C0C6  846F    STY  $6F
C0C8  A901    LDA  #$01
C0CA  8516    STA  $16
C0CC  68      PLA
C0CD  A8      TAY
C0CE  68      PLA
C0CF  AA      TAX
C0D0  68      PLA
C0D1  40      RTI
```

《大赛车》的操纵器采样是在中断处理中进行的，读出的各按键值存于零页单元$6B～$70中，其对应关系见表5－1。

表5－1 《大赛车》操纵器键码对照表

按键名称	$6B	$6C	$6D	$6E	$6F	$70
上	00	00	00	00	01	FF
下	00	00	00	00	00	FF
左	00	00	00	00	FF	01
右	00	00	00	00	FF	00
A	00	00	01	00	FF	FF
B	00	00	00	01	FF	FF
SELECT	01	00	00	00	FF	FF
START	00	01	00	00	FF	FF

程序No.5－29很好理解，不多赘述。

2、《小学算术一年级》的操纵器采样

《小学算术一年级》的操纵器采样程序为一独立的子程序，它能一次读出两个操纵器的按键代码。程序见No.5－30。

No.5－30《小学算术》操纵器采样程序

```
AEEE  A201     LDX  #$01
AEF0  8E1640   STX  $4016
AEF3  CA       DEX
AEF4  8E1640   STX  $4016
AEF7  20FBAE   JSR  $AEFB       ;读0#操纵器
AEFA  E8       INX              ;读1#操纵器
AEFB  A008     LDY  #$08
```

```
AEFD  48      PHA
AEFE  BD1640  LDA  $4016,X
AF01  851E    STA  $1E
AF03  4A      LSR
AF04  051E    ORA  $1E
AF06  4A      LSR
AF07  68      PLA
AF08  2A      ROL
AF09  88      DEY
AF0A  D0F1    BNE  $AEFD
AF0C  851E    STA  $1E
AF0E  B506    LDA  $06,X
AF10  49FF    EOR  #$FF
AF12  251E    AND  $1E
AF14  9508    STA  $08,X
AF16  A51E    LDA  $1E
AF18  9506    STA  $06,X
AF1A  60      RTS
```

程序 No. 5－30读出的按键代码存入＄06～＄09单元中，其中0＃操纵器代码存＄06、＄08；1＃操纵器代码存＄07、＄09。为什么一个操纵器代码存入两个单元中呢？分析一下 No. 5－30就可发现（以0＃操纵器为例），＄08单元存按键的瞬间值，即存按键脉冲上升沿，且只存一个中断周期，一个中断周期过去后其值又为零；而＄06单元虽然也存按键脉冲的上升沿，但可长期保存，只要不改按其它键则按键代码一直保持，直到改按新的按键为止。按键代码的对应关系见表5－2。

表5－2 《小学算术》操纵器按键代码表

按键	不按	上	下	左	右	A	B	SELECT	START
代码	00	08	04	02	01	80	40	20	10

3、《魂斗罗》的操纵器采样

《魂斗罗》游戏的操纵器采样处理也是一个独立的子程序，它被安排在中断处理中调用。采样中以＄1D单元为读0＃、1＃操纵器的转换开关，可一次读出两个操纵器的按键代码。程序见 No. 5－31。

No. 5－31《魂斗罗》操纵器采样程序

```
C35B  2097C3  JSR  $C397     ;操纵器采样子程序
C35E  A504    LDA  $04
C360  8500    STA  $00
C362  A505    LDA  $05
C364  8501    STA  $01
C366  2097C3  JSR  $C397     ;第二次采样认定
C369  A201    LDX  #$01
C36B  B504    LDA  $04,X
C36D  D500    CMP  $00,X
C36F  F004    BEQ  $C375
C371  B5F9    LDA  $F9,X
```

```
C373   9504     STA   $04,X
C375   CA       DEX
C376   10F3     BPL   $C36B
C378   A51D     LDA   $1D          ;由$1D控制转换
C37A   2904     AND   #$04
C37C   D006     BNE   $C384
C37E   A504     LDA   $04
C380   0505     ORA   $05
C382   8504     STA   $04
C384   A201     LDX   #$01
C386   B504     LDA   $04,X
C388   A8       TAY
C389   55F9     EOR   $F9,X
C38B   3504     AND   $04,X
C38D   95F5     STA   $F5,X
C38F   94F1     STY   $F1,X
C391   94F9     STY   $F9,X
C393   CA       DEX
C394   10F0     BPL   $C386
C396   60       RTS
C397   A201     LDX   #$01         ;采样子程序
C399   8E1640   STX   $4016
C39C   CA       DEX
C39D   8E1640   STX   $4016
C3A0   A008     LDY   #$08
C3A2   AD1640   LDA   $4016
C3A5   8507     STA   $07
C3A7   4A       LSR
C3A8   0507     ORA   $07
C3AA   4A       LSR
C3AB   2604     ROL   $04
C3AD   AD1740   LDA   $4017
C3B0   8507     STA   $07
C3B2   4A       LSR
C3B3   0507     ORA   $07
C3B5   4A       LSR
C3B6   2605     ROL   $05
C3B8   88       DEY
C3B9   D0E7     BNE   $C3A2
C3BB   60       RTS
```

程序 No. 5－31读出的按键代码与表5－2相同，读键过程留给读者分析。

4、BS. 2A 系统操纵器采样子程序的调用

在 F BASIC 指令中有两条操纵器采样指令：STICK、STRIG，这两条指令分别完成操纵器的方向键和功能键的采样。STICK 指令的返回值与上、下、左、右键的对应关系是8、4、2、1；

STRIG 指令的返回值与 A、B、SELECT、START 键的对应关系也是8、4、2、1。我们在进行机器语言编程时可以调用这两条指令的读键子程序。为便于其它系统的调用，特将该子程序列于 No. 5—32。

No. 5—32操纵器采样子程序

```
A682   A917      LDA   #$17
A684   AEA916    LDX   $16A9
A687   8519      STA   $19
A689   A940      LDA   #$40
A68B   851A      STA   $1A
A68D   A905      LDA   #$05
A68F   8D1640    STA   $4016
A692   EA        NOP
A693   EA        NOP
A694   A904      LDA   #$04
A696   8D1640    STA   $4016
A699   EA        NOP
A69A   EA        NOP
A69B   A208      LDX   #$08
A69D   A000      LDY   #$00
A69F   B119      LDA   ($19),Y
A6A1   4A        LSR
A6A2   2690      ROL   $90
A6A4   EA        NOP
A6A5   EA        NOP
A6A6   CA        DEX
A6A7   D0F6      BNE   $A69F
A6A9   60        RTS
```

No. 5—32的操纵器采样子程序可同时读出方向键、功能键的键值，其返回值存于 $ 90单元，按键与键值的对应关系同表5—2。

程序 No. 5—32的调用方法如下：

当对0#操纵器采样时要调 $ A685；

当对1#操纵器采样时要调 $ A682；

调用前要注意保护好 $ 19、$ 1A、$ 90单元的现场。

5·5·3 卡通的定义

卡通的定义过程就是把卡通块的四个参数(Y 座标、卡通图块代码、卡通的显示状态、X 座标)写入选定的卡通定义区。综合比较各种软件的卡通定义程序，唯有对话工作系统中的卡通定义程序独立性最强，最利于理解，最方便移植。因此，我们仅以它为例介绍卡通定义程序的编写。

对话工作系统的卡通定义过程共分两步完成：首先把卡通的显示状态送入定义区，并置 Y 座标为零；然后再把卡通的座标参数和图形代码依次送入卡通定义区。下面以例六为例讨论这一定义过程：

例六、用八个卡通块定义一个16×32点阵的卡通，即横向为两个卡通块、纵向为四个卡通块(《魂斗罗》游戏中的主人翁即为16×32点阵的卡通)。

一、卡通显示状态的定义

卡通显示状态的定义程序见 No. 5－33

No. 5－33卡通的显示状态定义子程序

```
EC86   A000     LDY  #$00
EC88   B100     LDA  ($00),Y
EC8A   C9FF     CMP  #$FF
EC8C   F013     BEQ  $ECA1      ;读数为 HFF 时结束定义
EC8E   8502     STA  $02        ;状态参数存入$02
EC90   C8       INY
EC91   B100     LDA  ($00),Y
EC93   8503     STA  $03        ;定义卡通块的数量存入$03
EC95   C8       INY
EC96   B100     LDA  ($00),Y
EC98   8504     STA  $04
EC9A   20A2EC   JSR  $ECA2      ;调状态定义子程序
EC9D   C8       INY
EC9E   4C88EC   JMP  $EC88
ECA1   60       RTS
```

定义卡通状态子程序

```
ECA2   98       TYA
ECA3   48       PHA
ECA4   A902     LDA  #$02
ECA6   8505     STA  $05
ECA8   A603     LDX  $03
ECAA   A000     LDY  #$00
ECAC   A900     LDA  #$00
ECAE   9104     STA  ($04),Y    ;对 Y 座标单元置零
ECB0   C8       INY
ECB1   C8       INY
ECB2   A502     LDA  $02
ECB4   9104     STA  ($04),Y    ;状态参数送入定义区
ECB6   C8       INY
ECB7   C8       INY
ECB8   CA       DEX
ECB9   D0F1     BNE  $ECAC
ECBB   68       PLA
ECBC   A8       TAY
ECBD   60       RTS
```

在调用 No. 5－33之前要先对$00、$01两单元置入数据区首址；对数据区置入状态参数。数据区的数据结构如下：每三个数据为一组，第一个数据为卡通状态；第二个数据为本次定义卡通块的数量；第三个数据为卡通块在卡通定义区的地址低位；最后还要置入定义结束标志

HFF。

对于例六，假如我们选定卡通的状态参数为03；其定义参数放在＄210～＄22F连续的H20个单元中，并选定数据区首址为＄7E00。那么，数据区结构如下：

7E00　03　08　10　FF

对＄00单元置入00、对＄01单元置入H7E，现在我们调用程序No. 5－33，分析卡通状态的定义过程：

执行＄EC86，从数据区读出卡通的状态参数存入＄02，有＄02＝03；读出定义卡通块的数量存入＄03，有＄03＝08；读出卡通定义区地址低位存入＄04，有＄04＝H10；

调＄ECA2，令＄05＝02，使＄04、＄05指向该卡通定义区的首址＄210；

执行＄ECAC，对Y座标单元置入零；

执行＄ECB2，把卡通的状态参数存入二页；

令X减一，X不等于零则返回＄ECAC，继续定义下一个卡通块的状态；

若X等于零，则结束状态定义，返回调用程序＄EC9A；

令Y加一，转＄EC88，继续从数据区读数，若读出的数据不等于HFF，则继续定义下一个卡通的状态，现在读出的数据是HFF，则结束定义。

通过调用程序No. 5－33，有关二页单元的内容如下：

＄210＝00	＄212＝03	＄214＝00	＄216＝03
＄218＝00	＄21A＝03	＄21C＝00	＄21E＝03
＄220＝00	＄222＝03	＄224＝00	＄226＝03
＄228＝00	＄22A＝03	＄22C＝00	＄22E＝03

2、卡通的座标、代码的定义

定义卡通的座标、代码子程序见No. 5－34：

No. 5－34定义卡通座标、代码子程序

```
ECBF  48    PHA
ECC0  98    TYA
ECC1  48    PHA
ECC2  A902  LDA  #$02
ECC4  8505  STA  $05
ECC6  A90F  LDA  #$0F
ECC8  2503  AND  $03
ECCA  8507  STA  $07       ;取$03单元的低位置入$07
ECCC  A503  LDA  $03
ECCE  4A    LSR
ECCF  4A    LSR
ECD0  4A    LSR
ECD1  4A    LSR
ECD2  8506  STA  $06       ;取$03单元的高位置入$06
ECD4  AA    TAX
ECD5  A900  LDA  #$00
ECD7  18    CLC
ECD8  6507  ADC  $07
ECDA  CA    DEX
```

```
ECDB  D0FB    BNE    $ECD8
ECDD  8508    STA    $08
ECDF  20EAEC  JSR    $ECEA       ;定义卡通的代码
ECE2  2005ED  JSR    $ED05       ;定义卡通的座标
ECE5  68      PLA
ECE6  A8      TAY
ECE7  68      PLA
ECE8  AA      TAX
ECE9  60      RTS
```

定义卡通的代码子程序

```
ECEA  A502    LDA    $02
ECEC  A608    LDX    $08
ECEE  A001    LDY    #$01
ECF0  9104    STA    ($04),Y     ;卡通的代码存入定义区
ECF2  18      CLC
ECF3  6901    ADC    #$01
ECF5  C8      INY
ECF6  48      PHA
ECF7  B104    LDA    ($04),Y
ECF9  293F    AND    #$3F
ECFB  9104    STA    ($04),Y     ;限定卡通的状态参数不大于H3F
ECFD  68      PLA
ECFE  C8      INY
ECFF  C8      INY
ED00  C8      INY
ED01  CA      DEX
ED02  D0EC    BNE    $ECF0
ED04  60      RTS
```

定义卡通的座标子程序

```
ED05  A000    LDY    #$00
ED07  A606    LDX    $06
ED09  A500    LDA    $00
ED0B  8509    STA    $09
ED0D  A501    LDA    $01
ED0F  9104    STA    ($04),Y     ;定义卡通的Y座标
ED11  C8      INY
ED12  C8      INY
ED13  C8      INY
ED14  A509    LDA    $09
ED16  9104    STA    ($04),Y     ;定义卡通的X座标
ED18  18      CLC
ED19  6908    ADC    #$08
ED1B  8509    STA    $09
ED1D  C8      INY
```

```
ED1E  CA    DEX
ED1F  D0EC  BNE   $ED0D
ED21  A501  LDA   $01
ED23  18    CLC
ED24  6908  ADC   #$08
ED26  8501  STA   $01
ED28  C607  DEC   $07
ED2A  D0DB  BNE   $ED07
ED2C  60    RTS
```

调用程序 No. 5－34之前，必须先对如下单元置入数据：

对$00单元置入卡通的 X 座标值；对$01单元置入卡通的 Y 座标值；

对$02单元置入第一个卡通块的代码(以后卡通块的代码则顺序置入)；

对$03单元置入卡通的定义指南，所谓卡通的定义指南的结构为：其高位标志着横排卡通块的数量，其低位标志着竖排卡通块的数量；

对$04单元置入卡通块定义区的地址低位；

对$05单元置入卡通块定义区的地址高位；

假定例六中卡通的 X 座标为 H60、Y 座标为 H90、第一个卡通块的代码为 HBC，那么$00～$05单元的数据如下：

$00=60　　$01=90　　$02=BC

$03=24　　$04=10　　$05=02

现在分析定义卡通座标、代码的过程：

执行$ECBE，把$03单元的高位02置入$06、低位04置入$07；

由$06控制$07的累加次数，将累加之和08置入$08，以控制定义卡通块的总数量；

调$ECEA，把卡通代码置入定义区，同时限定卡通的状态参数不大于 H3F；

调$ED05，把卡通的 Y 座标、X 座标参数置入定义区；

通过程序 No. 5－33、No. 5－34的处理，卡通定义区的有关单元置入了如下参数：

$210=90　　$214=90

$211=BC　　$215=BD

$212=03　　$216=03

$213=60　　$217=68

*　*　*　*　*

$218=98　　$21C=98

$219=BE　　$21D=BF

$21A=03　　$21E=03

$21B=60　　$21F=68

*　*　*　*　*

$220=A0　　$224=A0

$221=C0　　$225=C1

$222=03　　$226=03

$223=60　　$227=68

*　*　*　*　*

$228=A8　　　$22C=A8

$229=C2　　　$22D=C3

$22A=03　　　$22E=03

$22B=60　　　$22F=68

以上定义了在螃蟹背上立一个乌龟的16×32点阵的大卡通，由于卡通字库的限制，这仅仅是示意性的。

在实际应用中，我们必须周密的设计数据区，并配置一个根据按键的种类分别对$00～$05单元赋值的控制程序，以保证卡通按照操纵器的指令动作。

3、实用的卡通定义子程序

享有BS.2A版本F BASIC工作系统的朋友可以直接调用子程序$EC86、$ECBE，从而可极大的简化定义程序。没有BS.2A版本的朋友则只能根据No.5－33、No.5－34另编两段子程序供调用。实用的卡通定义子程序见No.5－35。

在编写定义卡通的子程序之前，要先选好一部分零页单元作为卡通参数存储器，选取原则是尽量避开系统的使用，选取闲置的单元。我们选取$50、$51、$77～$7B共七个零页单元为定义工作单元，它们的分工如下：

$50作为状态定义开关。一般情况下，卡通的状态以经定义后，一般不会经常改变，所以设置一个软开关以减少子程序的调用开销。

$51作为卡通状态定义数据区地址的偏移量，以便从数据区查出定义卡通状态的数据区首址存入$00、$01单元。

$77作为卡通X座标的存储器。

$78作为卡通Y座标的存储器。

$79作为卡通块的图形库代码存储器。

$7A作为一次定义卡通块的数量和排列形式。

$7B作为定义卡通块的二页地址低位。

No.5－35实用卡通定义子程序

```
7F00  20357F   JSR   $7F35     ;保护现场
7F03  A550     LDA   $50
7F05  F00F     BEQ   $7F16     ;检查卡通状态定义开关
7F07  A451     LDY   $51
7F09  B9807F   LDA   $7F80,Y
7F0C  8500     STA   $00
7F0E  B9817F   LDA   $7F81,Y
7F11  8501     STA   $01
7F13  2086EC   JSR   $EC86     ;定义卡通状态
7F16  A577     LDA   $77
7F18  8500     STA   $00
7F1A  A578     LDA   $78
7F1C  8501     STA   $01
7F1E  A579     LDA   $79
7F20  8502     STA   $02
7F22  A57A     LDA   $7A
```

```
7F24  8503     STA   $03
7F26  A57B     LDA   $7B
7F28  8504     STA   $04
7F2A  A902     LDA   #$02
7F2C  8505     STA   $05
7F2E  20BEEC   JSR   $ECBE
7F31  20437F   JSR   $7F43        ;恢复现场
7F34  60       RTS
      保护现场子程序
7F35  48       PHA
7F36  8A       TXA
7F37  48       PHA
7F38  98       TYA
7F39  48       PHA
7F3A  A209     LDX   #$09
7F3C  B500     LDA   $00,X
7F3E  48       PHA
7F3F  CA       DEX
7F40  10FA     BPL   $7F3C
7F42  60       RTS
      恢复现场子程序
7F43  A200     LDX   #$00
7F45  68       PLA
7F46  9500     STA   $00,X
7F48  E8       INX
7F49  E00A     CPX   #$0A
7F4B  90F8     BCC   $7F45
7F4D  68       PLA
7F4E  A8       TAY
7F4F  68       PLA
7F50  AA       TAX
7F51  68       PLA
7F52  60       RTS
```

程序 No. 5－35中选择了卡通状态数据区首址为＄7F80，用户可以自由选择。选定后只要更改＄7F09、＄7F0E 两程序行的数据地址就可以了。

5·5·4 《大赛车》游戏卡通定义程序简介

《大赛车》游戏中的主卡通——赛车，一般由十六至十八个卡通块组成(依据赛车直行和转弯的状态)。其定义程序为＄D2B2～＄D400、其数据区为＄DA13～＄DD0B，现简单介绍如下。程序见 No. 5－36。

No. 5－36赛车定义子程序

```
D2B2  A550     LDA   $50
D2B4  F00A     BEQ   $D2C0
```

```
D2B6  A551     LDA   $51
D2B8  C9A0     CMP   #$A0
D2BA  9004     BCC   $D2C0
D2BC  A901     LDA   #$01
D2BE  8575     STA   $75
D2C0  A556     LDA   $56
D2C2  F003     BEQ   $D2C7
D2C4  4C54F2   JMP   $F254     ;撞车时转$F254定义爆炸
D2C7  A55A     LDA   $5A
D2C9  F003     BEQ   $D2CE
D2CB  4CB5F3   JMP   $F3B5     ;游戏成功时定义赛车越去越远，
D2CE  A90B     LDA   #$0B      形状越来越小
D2D0  A470     LDY   $70       ;根据$70单元的数值调整$7C
D2D2  3016     BMI   $D2EA
D2D4  F00A     BEQ   $D2E0
D2D6  C67C     DEC   $7C
D2D8  A57C     LDA   $7C
D2DA  100E     BPL   $D2EA
D2DC  A900     LDA   #$00
D2DE  F00A     BEQ   $D2EA
D2E0  E67C     INC   $7C
D2E2  A57C     LDA   $7C
D2E4  C916     CMP   #$16
D2E6  9002     BCC   $D2EA
D2E8  A916     LDA   #$16
D2EA  857C     STA   $7C
D2EC  A550     LDA   $50       ;根据$50、$51调整$55、$54
D2EE  4A       LSR
D2EF  A551     LDA   $51
D2F1  6A       ROR
D2F2  18       CLC
D2F3  6555     ADC   $55
D2F5  8555     STA   $55
D2F7  9006     BCC   $D2FF
D2F9  A554     LDA   $54
D2FB  4905     EOR   #$05
D2FD  8554     STA   $54
D2FF  A47C     LDY   $7C       ;根据$7C的取值调整$19，然后
D301  B945DA   LDA   $DA45,Y   以$19为地址偏移量
D304  18       CLC
D305  6554     ADC   $54       ;对卡通数据地址存储单元$22、
D307  8519     STA   $19       $23赋值；
D309  A8       TAY
D30A  B913DA   LDA   $DA13,Y
```

```
D30D 8522    STA  $22
D30F B91DDA  LDA  $DA1D,Y
D312 8523    STA  $23
D314 B927DA  LDA  $DA27,Y  ;对同时送二页、三页定义卡通块的
D317 8517    STA  $17      数量存储单元$17赋值
D319 B931DA  LDA  $DA31,Y  ;对只送三页卡通数量存储单元$18赋值
D31C 8518    STA  $18
D31E B93BDA  LDA  $DA3B,Y  ;对只送二页卡通数量存储单元$1A赋值
D321 851A    STA  $1A
D323 A208    LDX  #$08
D325 A55C    LDA  $5C
D327 F052    BEQ  $D37B
D329 A57B    LDA  $7B      ;$5C≠0时定义刹车时轮胎后的
D32B 4901    EOR  #$01     飞溅尘土
D32D 857B    STA  $7B
D32F A9D2    LDA  #$D2
D331 2008D1  JSR  $D108
D334 A57B    LDA  $7B
D336 F002    BEQ  $D33A
D338 A940    LDA  #$40
D33A A20A    LDX  #$0A
D33C 2008D1  JSR  $D108
D33F A57B    LDA  $7B
D341 D007    BNE  $D34A
D343 A966    LDA  #$66
D345 A26E    LDX  #$6E
D347 4C4ED3  JMP  $D34E
D34A A992    LDA  #$92
D34C A28A    LDX  #$8A
D34E 8D0B02  STA  $020B
D351 8D0B03  STA  $030B
D354 8E0F02  STX  $020F
D357 8E0F03  STX  $030F
D35A A55D    LDA  $5D
D35C A8      TAY
D35D 0A      ASL
D35E 18      CLC
D35F 69F3    ADC  #$F3
D361 8D0902  STA  $0209
D364 8D0903  STA  $0309
D367 6901    ADC  #$01
D369 8D0D02  STA  $020D
D36C 8D0D03  STA  $030D
D36F A57B    LDA  $7B
```

```
D371  D006     BNE   $D379
D373  C8       INY
D374  98       TYA
D375  2903     AND   #$03
D377  855D     STA   $5D
D379  A210     LDX   #$10
D37B  A000     LDY   #$00          ;定义赛车
D37D  B122     LDA   ($22),Y
D37F  18       CLC
D380  69B8     ADC   #$B8
D382  9D0002   STA   $0200,X
D385  9D0003   STA   $0300,X
D388  E8       INX
D389  C8       INY
D38A  B122     LDA   ($22),Y
D38C  9D0002   STA   $0200,X
D38F  9D0003   STA   $0300,X
D392  E8       INX
D393  C8       INY
D394  B122     LDA   ($22),Y
D396  9D0002   STA   $0200,X
D399  9D0003   STA   $0300,X
D39C  E8       INX
D39D  C8       INY
D39E  B122     LDA   ($22),Y
D3A0  18       CLC
D3A1  6980     ADC   #$80
D3A3  9D0002   STA   $0200,X
D3A6  9D0003   STA   $0300,X
D3A9  E8       INX
D3AA  C8       INY
D3AB  C417     CPY   $17
D3AD  90CE     BCC   $D37D
D3AF  862B     STX   $2B
D3B1  B122     LDA   ($22),Y       ;定义卡通数据只送二页的处理
D3B3  18       CLC
D3B4  69B8     ADC   #$B8
D3B6  9D0002   STA   $0200,X
D3B9  E8       INX
D3BA  C8       INY
D3BB  B122     LDA   ($22),Y
D3BD  9D0002   STA   $0200,X
D3C0  E8       INX
D3C1  C8       INY
```

```
D3C2  B122      LDA   ($22),Y
D3C4  9D0002    STA   $0200,X
D3C7  E8        INX
D3C8  C8        INY
D3C9  B122      LDA   ($22),Y
D3CB  18        CLC
D3CC  6980      ADC   #$80
D3CE  9D0002    STA   $0200,X
D3D1  E8        INX
D3D2  C8        INY
D3D3  C41A      CPY   $1A
D3D5  90DA      BCC   $D3B1
D3D7  A62B      LDX   $2B          ;定义卡通的数据只送三页的处理
D3D9  B122      LDA   ($22),Y
D3DB  18        CLC
D3DC  69B8      ADC   #$B8
D3DE  9D0003    STA   $0300,X
D3E1  E8        INX
D3E2  C8        INY
D3E3  B122      LDA   ($22),Y
D3E5  9D0003    STA   $0300,X
D3E8  E8        INX
D3E9  C8        INY
D3EA  B122      LDA   ($22),Y
D3EC  9D0003    STA   $0300,X
D3EF  E8        INX
D3F0  C8        INY
D3F1  B122      LDA   ($22),Y
D3F3  18        CLC
D3F4  6980      ADC   #$80
D3F6  9D0003    STA   $0300,X
D3F9  E8        INX
D3FA  C8        INY
D3FB  C418      CPY   $18
D3FD  90DA      BCC   $D3D9
D3FF  862B      STX   $2B
D341  60        RTS
```

定义卡通赛车的数据

以下由$DA13～$DA26组成十组数据地址用于送入$22、$23，随后的十个数据送入$17、十个数据送入$18、十个数据送入$1A。

```
DA13           5C A4 E8 28 6C B4 FC 40 80 C4 DA DA DA
DA20  DB DB DB DB DC DC DC 30 2C 30 2C 30 30 2C 30 2C
DA30  30 48 44 40 44 48 48 44 40 44 48 3C 38 38 38 3C
DA40  3C 38 38 38 3C 00 01 01 01 01 01 01 01 01 01 01
```

DA50 02 03 03 03 03 03 03 03 03 03 03 04

以下为定义赛车直行、左转、右转共十种形状的数据

DA5C 00 1F 40 E8
DA60 00 1D 40 F8 00 1C 40 00 08 23 40 F0 08 22 40 F8
DA70 08 21 40 00 10 28 40 F8 10 27 40 00 10 26 40 08
DA80 18 2D 40 F8 18 2C 40 00 18 2B 40 08 08 20 40 08
DA90 10 25 40 10 18 2A 40 10 08 24 40 E8 10 29 40 F0
DAA0 18 2E 40 F0 00 00 40 EE 00 0A 40 06 08 11 40 F6
DAB0 08 10 40 FE 08 0F 40 06 10 16 40 F6 10 15 40 FE
DAC0 10 14 40 06 18 1A 40 F6 18 19 40 FE 18 18 40 06
DAD0 08 0E 40 0E 10 13 40 0E 18 07 40 0E 08 12 40 EE
DAE0 10 17 40 EE 18 1B 40 EE 00 00 00 F0 00 00 40 08
DAF0 08 02 00 F0 08 03 00 F8 08 03 40 00 08 02 40 08
DB00 10 05 00 F4 10 06 00 FC 10 05 40 04 18 08 00 F4
DB10 18 09 00 FC 18 08 40 04 10 04 00 EC 18 07 00 EC
DB20 10 04 40 0C 18 07 40 0C 00 0A 00 F2 00 00 00 0A
DB30 08 0F 00 F2 08 10 00 FA 08 11 00 02 10 14 00 F2
DB40 10 15 00 FA 10 16 00 02 18 18 00 F2 18 19 00 FA
DB50 18 1A 00 02 08 0E 00 EA 10 13 00 EA 18 07 00 EA
DB60 08 12 00 0A 10 17 00 0A 18 1B 00 0A 00 1C 00 F8
DB70 00 1D 00 00 00 1F 00 10 08 21 00 F8 08 22 00 00
DB80 08 23 00 08 10 26 00 F0 10 27 00 F8 10 28 00 00
DB90 18 2B 00 F0 18 2C 00 F8 18 2D 00 00 08 20 00 F0
DBA0 10 25 00 E8 18 2A 00 E8 08 24 00 10 10 29 00 08
DBB0 18 2E 00 08 00 3B 40 E8 00 1D 40 F8 00 1E 40 00
DBC0 08 23 40 F0 08 22 40 F8 08 3A 40 00 10 3E 40 F8
DBD0 10 27 40 00 10 26 40 08 18 2D 40 F8 18 2C 40 00
DBE0 18 2B 40 08 08 20 40 08 10 3D 40 10 18 2A 40 10
DBF0 08 3C 40 E8 10 3F 40 F0 18 2E 40 F0 00 00 40 EE
DC00 00 0A 40 06 08 11 40 F6 08 10 40 FE 08 33 40 06
DC10 10 37 40 F6 10 15 40 FE 10 14 40 06 18 1A 40 F6
DC20 18 19 40 FE 18 18 40 06 08 0E 40 0E 10 36 40 0E
DC30 18 07 40 0E 08 35 40 EE 10 38 40 EE 18 1B 40 EE
DC40 00 00 00 F0 00 00 40 08 08 30 00 F0 08 03 00 F8
DC50 08 03 40 00 08 30 40 08 10 05 00 F4 10 06 00 FC
DC60 10 05 40 04 18 08 00 F4 18 09 00 FC 18 08 40 04
DC70 10 31 00 EC 18 07 00 EC 10 31 40 0C 18 07 40 0C
DC80 00 0A 00 F2 00 00 00 0A 08 33 00 F2 08 10 00 FA
DC90 08 11 00 02 10 14 00 F2 10 15 00 FA 10 37 00 02
DCA0 18 18 00 F2 18 19 00 FA 18 1A 00 02 08 0E 00 EA
DCB0 10 36 00 EA 18 07 00 EA 08 35 00 0A 10 38 00 0A
DCC0 18 1B 00 0A 00 1E 00 F8 00 1D 00 00 00 3B 00 10
DCD0 08 3A 00 F8 08 22 00 00 08 23 00 08 10 26 00 F0
DCE0 10 27 00 F8 10 3E 00 00 18 2B 00 F0 18 2C 00 F8

```
DCF0  18 2D 00 00 08 20 00 F0 10 3D 00 E8 18 2A 00 E8
DD00  08 3C 00 10 10 3F 00 08 18 2E 00 08
```

程序 No.5－36仅仅是赛车正常行驶状态的定义处理，并不包括其它干扰赛车、路标、指示标语、撞车爆炸、赛车越远越小等卡通的定义，也不包括撞车、下路等情况的判断处理。由此可见，在任天堂游戏中卡通的控制处理是比较复杂的。由于 No.5－36是从软件中截取的一段，前后连贯性较差，故仅作简单的介绍，旨在使朋友们初步接触一下实实在在的任天堂游戏，对它有一个初步的认识。详细的讨论将在《任天堂游戏编程探密》中进行。

5·6　音响系统的编程技巧

音响——是任天堂游戏的三大支柱之一。一个游戏的功能评价高低，音响也是决定因素之一。《魂斗罗》之所以倍受青睐，虽然主要取决于游戏结构的新颖和神奇的科幻构思，但也与它那绝妙的效果音响和恰如其分的背景音乐伴奏密切相关的。以上我们曾讨论了音响系统发声的基本原理，本节重点讨论系统效果音响、背景音乐的调用方法和背景音乐旋律演奏程序的编写技巧。

5·6·1　系统效果音响的调用

通过第三章的讨论，我们已经了解到人机对话工作系统中有着丰富的效果音响，若能在我们编写的程序中直接调用，必能使编程质量提高一个档次。从表3－32可见，对话系统中由零页单元＄FC 控制发出的效果音有四种、由＄FE 控制发出的效果音有五种、由＄FF 控制发出的效果音有四种，共有十三种。若对＄FC 置入 H21、H22……，还可以发出几十种不同的背景噪声，可用以模仿风声、雨声、引擎声等等。

程序 No.5－37 可以实现上述系统效果音响的调用。

No.5－37 系统音响调用子程序

```
7EF7  A900     LDA   #$00
7EF9  85EE     STA   $EE
7EFB  A97F     LDA   #$7F
7EFD  85EF     STA   $EF
7EFF  60       RTS
7F00  48       PHA
7F01  8A       TXA
7F02  48       PHA
7F03  98       TYA
7F04  48       PHA
7F05  205BFD   JSR   $FD5B
7F08  A900     LDA   #$00
7F0A  85FE     STA   $FE
7F0C  85FF     STA   $FF
7F0E  68       PLA
7F0F  A8       TAY
7F10  68       PLA
7F11  AA       TAX
```

```
7F12  68        PLA
7F13  4C0F88    JMP    $880F
```

输入程序 No. 5－37，执行“CALL &7EF7”命令后，F BASIC 工作系统已作好了发声准备，现在只要向 $FC、$FE、$FF 单元置入适当的参数，就可发出希望的效果音响。

在《电脑游戏机 F BASIC 语言与编程技巧》一书中，我们曾讨论了金页古塔搬移的演示程序，那个程序仅模拟了古塔搬移过程，效果较一般。现在我们给它加入效果音响，并增加了操纵器选项处理。朋友们再体会一次古塔搬移过程，肯定会有一个全新的感觉。可见，由于有了音响的衬托，则使一般化的程序产生了游戏的韵味。改造后的金页古塔搬移程序见 No. 5－38。

```
5 REM No. 5－38金页古塔搬移动态演示
10 CGSET 0, 0 : CLS
20 LOCATE 5,3 : PRINT "TOWER TIER 3 4 5 6 7"
30 X1=16 : LOCATE X1,4 : PRINT CHR$(232)
40 PAUSE 10 : T=STICK(0) : F=153 : X2=X1 :IF T>2 THEN 40
50 IF T=1 AND X1<24 THEN GOSUB 440 : X1=X1+2
60 IF T=2 AND X1>16 THEN GOSUB 440 : X1=X1-2
70 LOCATE X2,4 : PRINT " " : LOCATE X1,4 :PRINT CHR$(232)
80 IF STRIG(0)=1 THEN D=VAL(SCR$(X1, 3)): F=250:GOSUB440 : GOTO 100
90 GOTO 40
100 DIM D$(D),D(D-1,2) : D$(0)="        "(八个空格)
110 FOR I=1 TO D : D$(I)=CHR$(196) : FOR J=0 TO I-1:D$(I)=D$(I)+CHR$(197): NEXT:
    D$(I)=MID$(D$(I),1,I)+CHR$(198) : NEXT
120 FOR I=0 TO D-1 : D(I,0)=I+1 : NEXT
130 CLS : GOSUB 410 : FOR I=1 TO 3 : LOCATE I*9-6,21:PRINT I : NEXT : X1=3 : LOCATE
    X1, 21 :PRINT CHR$(215)
140 LOCATE 7,4 : PRINT "GAME START" : PAUSE 25 :LOCATE 7,4 : PRINT D$(0); D$(0) : LO-
    CATE 9, 3 :PRINT "0 STEP"
150 GOSUB 430 : GOSUB 340 : GOSUB 430 : K1=K-1 :PRINT K" TO"; : GOSUB 340 : K2=K-1 :
    LOCATE 14,7 :PRINT K : I0=0
160 IF I0=D THEN F=146 : GOSUB 440 : GOTO 150
170 IF D(I0, K1)=0 THEN I0=I0+1 : GOTO 160
180 M1=D(I0, K1) : I1=0
190 IF I1=D THEN 220
200 IF D(I1, K2)=0 THEN I1=I1+1 : GOTO 190
210 IF D(I1, K2)<=M1 THEN F=146 : GOSUB 440 : GOTO 150
220 D(I0, K1)=0: D(I1-1, K2)=M1
230 GOSUB 410 : ST=ST+1 : LOCATE 8, 3 : PRINT ST" STEPS" : GOSUB 430 : I0=0 : I1=0
240 IF I0=2 THEN 290
250 IF I1=D THEN 290
260 IF I1+1=D(I1, I0+1) THEN I1=I1+1 : GOTO 250
270 I0=I0+1 : IF I0=2 THEN 150
280 I1=0 : GOTO 250
290 F=200 : FOR I=1 TO D : READ S2 : F=F+1 :GOSUB 440 : NEXT: LOCATE 4, 8: PRINT "
    GAME END":LOCATE 14,8 : PRINT " SCORE " S2*100/ST
```

```
300 LOCATE 3,1 0: PRINT "RE PLAY PRESS <START>"
310 IF STRIG(0)=1 THEN F=252 : GOSUB 440 : RUN
320 IF STRIG(0)=2 THEN END
330 GOTO 310
340 PAUSE 10 : T=STICK(0) : F=153 : X2=X1 :IF T>2 THEN 340
350 IF T=1 AND X1<21 THEN X1=X1+9 : GOSUB 440
360 IF T=2 AND X1>9 THEN X1=X1-9 : GOSUB 440
370 LOCATE X2,21 : PRINT " " : LOCATE X1, 21 :PRINT CHR$(215)
380 G=STRIG(0) : IF G<8 THEN 340
390 K=-(X1=3)-(X1=12)*2-(X1=21)*3 : F=250 :GOSUB 440
400 RETURN
410 FOR II=0 TO 2 : FOR I=0 TO D-1 : LOCATE II*9,21-D+I : PRINT D$(D(I,II)) : NEXT :
    NEXT : RETURN
420 DATA 1,3,7,15,31,63,127
430 LOCATE 9,7 : PRINT D$(0) : LOCATE 9,7 : RETURN
440 POKE 16405,7 : POKE 16384,31,F,239,8,: RETURN
```

5·6·2 系统背景音乐的调用

在对话工作系统的计算板、打字板工作时都有不同的背景音乐伴随始终，从而打破了寂寞、活跃了版面、提高了趣味。同时我们还知道，对话系统中还有两段暗藏的背景音乐没有使用。下面通过八皇后的动态演式改造程序 No. 5－39，介绍系统背景音乐的调用方法。

当输入 No. 5－37、执行了"CALL &7EF7"之后，F BASIC 系统不但具备了调用系统效果音响的功能，同时也具备了调用系统背景音乐的功能。所以，我们可以直接使用"POKE &FC，X"命令调用系统背景音乐。因此，我们可在程序 No. 5－39中直接调用。

```
5 REM No. 5-39动态演示八皇后
POKE 32736,72,138,72,152,72,169,255,141,21,64,32,91,253,104,168,104,170,104,76,15,136,169,
     224,133,238,169,127,133, 239,96
CALL 32575
```

以上两行程序为立即执行型，不加行号，键入程序后，回车即可。

```
10 DIM C(8), T(8), S(15), P(15) : D=1: POKE 252, 1
20 SPRITE ON : VIEW : GOSUB 500
30 POKE 252, 3
40 FOR I=0 TO 7 : DEF MOVE(I)=SPRITE(1,3,1,1,0,2): POSITION I,20,30+I*24 : MOVE I
50 NEXT
60 K=1
70 IF T(K)=1 OR S(8+D-K)=1 OR P(D+K-1)=1 THEN 110
80 C(D)=K : T(K)=1 : S(8+D-K)=1 : P(D+K-1)=1 :GOSUB 140
90 D=D+1 : IF D>8 THEN TA=TA+1 : LOCATE 24, 18 :PRINT TA; : POKE 252,1,0,0,64 : A$=
   INKEY$(0) :POKE 252,RND(2)+2 : GOTO 120
100 GOTO 60
110 K=K+1 : IF K<=8 THEN 70
120 D=D-1 : IF D<1 THEN SPRITE OFF : CLS :PRINT "TOTAL="; TA : POKE 254,32 : PAUSE
    500 : POKE 252,0,0,0 : END
```

```
130 K=C(D) : T(K)=0 : S(8+D-K)=0 : P(D+K-1)=0 :GOTO 110
140 X=20+(K-1)*24 : D1=D-1 : Y=30+D1*24
150 X0=XPOS(D1) : Y0=YPOS(D1)
160 IF X>X0 THEN DD=3 : SS=(X-X0)/2 : GOTO 180
170 DD=7 : SS=(X0-X)/2
180 DEF MOVE(D1)=SPRITE(1,DD,1,SS,0,2) :POSITION D1,X0,Y0 : MOVE D1
190 RETURN
500 FOR I=1 TO 3 : A$=A$+CHR$(255) : B$=B$+CHR$(253) : NEXT
510 FOR I=1 TO 8 : FOR K=1 TO 3
520 FOR J=1 TO 4 : PRINT A$;B$; : NEXT
530 IF I=8 AND K=3 THEN 550
540 PRINT : NEXT : SWAP A$,B$
550 NEXT
560 RETURN
```

程序 No. 5－39运行后，在轻松愉快的音乐伴奏下，八位皇后婷婷玉立在8×8格的黑白棋盘方阵中。现在你就是指挥布阵的领兵大元帅，三军鹄立，只待你一声令下。按任一键后，音乐节奏陡转张紧急促，阵地上，八个皇后瞻前顾后，穿梭奔跑，各自寻找自己的恰当位置。当八个皇后的位置全部达到互不侵犯的要求后，则静立不动，等待你检阅阵容，同时音乐节奏又转缓婉流畅。身临其境，你可有凛凛然的感受吗？

5·6·3 音乐旋律演奏程序的编写

以上通过调用系统发声子程序，收到了较满意的效果。但是，吃嚼过的甘蔗总觉得没滋味，可能许多朋友想尝试自己动手编写背景音乐程序。实际上，通过以上的讨论，朋友们应该能够编出这一程序，但考虑到在其它系统中编写背景音乐的因素，更为入门较晚的朋友也能早日登堂入室，特讨论如下：

我们仅以流行歌曲《小秘密》为例，具体介绍音乐旋律演奏程序的编写过程。

一、选取演奏声部及程序格式

演奏主旋律选择第一、第二主声部都可以，本例我们选择第二主声部。程序格式以$FC控制演奏的程序为参考加以改造（这样选择的目的：一方面使程序便于理解；另一方面便于加深对系统发声处理方法的认识。）。

二、确定程序区和数据区

$FC控制发声的数据区有三段：分为音量数据、音长数据和音符数据。由于音量、音长数据区较小，故选择$7FE0～$7FEF为音量数据区、$7FF0～$7FFF为音长数据区；由于《小秘密》的音符数据量不大，故选择$7E00为存放首址；选择$7F00为程序区首址；选择$7EF7～$7EFF为进入中断程序区（该段程序运行后可以删除）。

三、编写演奏控制程序

根据以上的选定，改写程序见 No. 5－40。

No. 5－40音乐旋律演奏子程序

```
7F00  A5FC    LDA   $FC
7F02  D006    BNE   $7F0A
7F04  85FB    STA   $FB
7F06  85F3    STA   $F3
```

```
7F08  F037     BEQ   $ 7F41
7F0A  45FB     EOR   $ FB
7F0C  F00A     BEQ   $ 7F18
7F0E  A5FC     LDA   $ FC
7F10  85FB     STA   $ FB
7F12  A900     LDA   # $ 00
7F14  85F4     STA   $ F4
7F16  F004     BEQ   $ 7F1C
7F18  C6F6     DEC   $ F6
7F1A  D025     BNE   $ 7F41
7F1C  A4F4     LDY   $ F4
7F1E  E6F4     INC   $ F4
7F20  B9007E   LDA   $ 7E00,Y
7F23  F0ED     BEQ   $ 7F12
7F25  AA       TAX
7F26  6A       ROR
7F27  8A       TXA
7F28  2A       ROL
7F29  2A       ROL
7F2A  2A       ROL
7F2B  2907     AND   # $ 07
7F2D  A8       TAY
7F2E  B9F07F   LDA   $ 7FF0,Y
7F31  85F6     STA   $ F6
7F33  8A       TXA
7F34  293E     AND   # $ 3E
7F36  C902     CMP   # $ 02
7F38  F007     BEQ   $ 7F41
7F3A  208A7F   JSR   $ 7F8A
7F3D  A900     LDA   # $ 00
7F3F  85F8     STA   $ F8
7F41  A4F8     LDY   $ F8
7F43  C008     CPY   # $ 08
7F45  B008     BCS   $ 7F4F
7F47  B9E07F   LDA   $ 7FE0,Y
7F4A  8D0440   STA   $ 4004
7F4D  E6F8     INC   $ F8
7F4F  60       RTS
```

四、写出乐谱

《小秘密》的简谱如下：

《小秘密》

```
4/4
‖:3 32 3 32 | 3 2 1 · 10 | 2 21 2 35 | 3 · 0  0  0 |
| 3 32 3 32 | 1 1 6 60  |02 23 2 1 6| 6 2 2 — — |
| 3 32 3 32 | 3 2 1 · 10 | 2 21 2 35 | 3 · 0 0 0  |
| 3 32 3 32 | 1 1 6 · 60 | 7 7 6 5 6 | 1 — — — |
| 05 55 5 35| 6 3 3 · 3  | 6 — 7 · 6| 5 5 3 2 2  |
| 0 3 3 5 6 | 3 21 6 —  |05 56 5 3 2| 1 — — — :‖
```

五、编写数据区

根据《小秘密》的节奏特点，若选取八个中断周期为八分音符，那么付点八分音符则为十二个中断周期，相应地四分音符为十六个中断周期，其它依次类推。故音长数据区如下：

7FF0	08	0C	10	18	20	28	30	40
拍节：	1/2	3/4	1	3/2	2	5/2	3	4

音色音量数据可根据选定的乐器类型选取。我们仍选取打击乐器音色，故编写音色音量数据区如下：

7FE0　　B5 B4 B3 B2 B1 B1 B1 B0

程序 No. 5－40的音符数据要求如下：

由数据的 D0、D7、D6三位依次排成二进制的三位数作为音符音长值的地址偏移量；由数据的D5、D4、D3、D2、D1、D0六位依次排成二进制的六位数(D0位恒为零)作为音符音调值的地址偏移量。据此编写音符数据区如下：

```
7E00    A0 20 1C A0 20 1C A0 1C 19 02 9C 1C 18 9C 20 26
7E10    E0 43 A0 20 1C A0 20 1C 98 18 D2 02 02 1C 1C 20
7E20    1C 98 92 9D A0 20 1C A0 20 1C A0 1C 19 02 9C 1C
7E30    18 9C 20 26 E0 43 A0 20 1C A0 20 1C 98 18 D2 02
7E40    96 96 12 8E 12 D9 02 26 26 26 A6 20 26 AA 20 61
7E50    2B EE 2A 26 A6 A0 DC 02 A0 20 A6 AA A0 1C 18 13
7E60    02 26 26 2A A6 A0 9C D9 00 00 00 00 0F 0F 0F 0F
```

六、编写音调单元赋值子程序

系统中原有的赋值子程序＄FC69～＄FC97已不适用乐曲《小秘密》的赋值需要，因此需要重新编写赋值子程序。为适用各声部赋值，仍采用原格式编写。程序见 No. 5－41。

No. 5－41音调单元赋值子程序

```
7F70    A07F    LDY   #$7F
7F72    8E0040  STX   $4000
7F75    8C0140  STY   $4001
7F78    A200    LDX   #$00
7F7A    A8      TAY
7F7B    B91E80  LDA   $801E,Y
7F7E    9D0240  STA   $4002,X
7F81    B91F80  LDA   $801F,Y
```

```
7F84    0908    ORA  #$08
7F86    9D0340  STA  $4003,X
7F89    60      RTS
7F8A    A07F    LDY  #$7F
7F8C    8C0540  STY  $4005
7F8F    A204    LDX  #$04
7F91    D0E7    BNE  $7F7A
7F93    A208    LDX  #$08
7F95    C902    CMP  #$02
7F97    F0F0    BEQ  $7F89
7F99    D0DF    BNE  $7F7A
```

七、编写调用程序

对于演奏子程序的调用程序，我们并不陌生，故直接写出程序。见 No. 5－42。

No. 5－42演奏子程序的调用程序

```
7F50    48      PHA
7F51    8A      TXA
7F52    48      PHA
7F53    98      TYA
7F54    48      PHA
7F55    A90F    LDA  #$0F
7F57    8D1540  STA  $4015
7F5A    20007F  JSR  $7F00
7F5D    EA      NOP
7F5E    EA      NOP
7F5F    EA      NOP
7F60    EA      NOP
7F61    EA      NOP
7F62    EA      NOP
7F63    68      PLA
7F64    A8      TAY
7F65    68      PLA
7F66    AA      TAX
7F67    68      PLA
7F68    4C0F88  JMP  $880F
```

程序 No. 5－42中留出了＄7F5D～＄7F62六个单元，以供调用两个其它声部的演奏子程序。

现在，万事具备，只要设置好中断程序入口，就可进行背景音乐的演奏了。显而易见，中断程序入口地址为＄7F50。故编写中断入口地址设置子程序如下：

```
7EF7    A950    LDA  #$50
7EF9    85EE    STA  $EE
7EFB    A97F    LDA  #$7F
7EFD    85EF    STA  $EF
7EFF    60      RTS
```

八、运行程序

把以上各程序写入各自的地址后，执行“CALL &7EF7”、再执行“POKE &FC，1”，就可以欣赏自己设计的背景音乐了。在自己设计的音乐伴奏下，操作电脑将是多么惬意啊！

九、演奏程序的完善

一个声部演奏总显单调，若再加入和声伴奏及打击乐节奏，效果将会更好。和声伴奏可以使用第一主声部，打击乐节奏可以使用第三主声部。下面仅介绍打击乐节奏的编程方法，和声伴奏留给读者练习。

为了使节奏感更强烈，且节奏的强弱随主旋律的周期而改变，特把用于第三主声部的《小秘密》歌曲的乐谱改写如下：

用于第三主声部的《小秘密》乐谱

4/4　　（注——每个音符都是1/2拍）

```
‖3 3 3 2 3 3 3 2|3 3 2 1 1 1 1 1|2 2 2 1 2 2 3 5|3 3 3 3 3 3 3 3|
|3 3 3 2 3 3 3 2|1 1 1 6 6 6 6 6|2 2 2 3 2 1 1 6|6 2 2 2 2 2 2 2|
|3 3 3 2 3 3 3 2|3 3 2 1 1 1 1 1|2 2 2 1 2 2 3 5|3 3 3 3 3 3 3 3|
|3 3 3 2 3 3 3 2|1 1 1 6 6 6 6 6|7 7 7 7 6 5 5 6|1 1 1 1 1 1 1 1|
|5 5 5 5 5 5 3 5|6 6 3 3 3 3 3 3|6 6 6 6 7 7 7 6|5 5 5 3 3 2 2 2|
|3 3 3 3 5 5 6 6|3 3 2 1 6 6 6 6|5 5 5 6 5 5 3 3 2 2|1111 1 1 1 0‖
```

节奏演奏程序见 No. 5－43。

No. 5－43第三主声部演奏子程序

```
7F9B    A5FA     LDA   $FA
7F9D    D00F     BNE   $7FAE
7F9F    A5FC     LDA   $FC
7FA1    85FA     STA   $FA
7FA3    D001     BNE   $7FA6
7FA5    60       RTS
7FA6    A900     LDA   #$00
7FA8    85F5     STA   $F5
7FAA    A901     LDA   #$01
7FAC    85F7     STA   $F7
7FAE    C6F7     DEC   $F7
7FB0    F001     BEQ   $7FB3
7FB2    60       RTS
7FB3    A4F5     LDY   $F5
7FB5    E6F5     INC   $F5
7FB7    B9007D   LDA   $7D00,Y
7FBA    D006     BNE   $7FC2
7FBC    85FA     STA   $FA
7FBE    8D0840   STA   $4008
7FC1    60       RTS
7FC2    20937F   JSR   $7F93
7FC5    A901     LDA   #$01
7FC7    8D0840   STA   $4008
```

```
7FCA        A908   LDA   # $08
7FCC        85F7   STA   $F7
7FCE        60     RTS
```

根据 No. 5－43对发声数据的要求，编写节奏演奏数据如下：

```
7D00    20 20 20 1C 20 20 20 1C 20 20 1C 18 18 18 18 18
7D10    1C 1C 1C 18 1C 1C 20 26 20 20 20 20 20 20 20 20
7D20    20 20 20 1C 20 20 20 1C 18 18 18 12 12 12 12 12
7D30    1C 1C 1C 20 1C 18 18 12 12 1C 1C 1C 1C 1C 1C 1C
7D40    20 20 20 1C 20 20 20 1C 20 20 1C 18 18 18 18 18
7D50    1C 1C 1C 18 1C 1C 20 26 20 20 20 20 20 20 20 20
7D60    20 20 20 1C 20 20 20 1C 18 18 18 12 12 12 12 12
7D70    16 16 16 16 12 0E 0E 12 18 18 18 18 18 18 18 18
7D80    26 26 26 26 26 26 20 26 2A 2A 20 20 20 20 20 20
7D90    2A 2A 2A 2A 2E 2E 2E 2A 26 26 26 20 20 1C 1C 1C
7DA0    20 20 20 20 26 26 2A 2A 20 20 1C 18 12 12 12 12
7DB0    26 26 26 2A 26 26 20 20 1C 1C 18 18 18 18 18 18
7DC0    18 00
```

把程序 No. 5－43和数据写入内存后，要在主控程序 No. 5－42中加一条调用子程序语句“JSR $7F9B”，其地址为：

```
7F5D  20 9B 7F
```

现在运行程序将会聆听到富有节奏感的背景音乐了。

读者可以调整两个声部的音色数据，以演奏自己喜爱的乐器效果。这些改造较简单，不多赘述。

第六章　实用程序专辑

本章重点介绍几个实用的工具程序，供朋友们在编程中调用。

6·1　6527 工作系统的小监控程序

由于目前流行的电脑游戏机都没有配备监控系统，故对内存单元的读、写操作只能靠编写 F BASIC 程序实现。这种方法虽然行之有效但效率较低。本节提供的小监控程序可以查看任意内存单元的内容，可以向任意 RAM 单元写入十六进制数据，使用方法与一般监控系统相同。具体使用方法如下：

一、进入监控

执行“CALL &7800”，就可由 F BASIC 工作状态进入小监控状态。

二、查看内存

查看内存格式为：

“＊■XXXX.YYYY”

——星号为小监控系统提示符；

——■为光标；

——XXXX 为查看内存的十六进制起始地址(1～4 位)；

——YYYY 为结束地址；

——“.”为查内存命令。

三、写入内存

写入内存格式为：

“＊■XXXX:YY YY YY YY YY YY ......”

——XXXX 为写内存的十六进制首地址；

——YY 为写入内存的十六进制数据，可一次写入八十个数据；

——“:”为写入内存命令。

四、退出监控系统

退出监控系统时只要键入“Q”后回车，就可返回 F BASIC 工作状态。

由于该监控系统只能读、写内存，故称之为小监控。程序清单见 No. 6－1。

No. 6－1 6527 小监控系统

```
7800    2007AC    JSR    $AC07    ;令光标下移一行，位于行首
7803    A92A      LDA    #$2A
7805    2034AC    JSR    $AC34    ;送显监控系统提示符“＊”
7808    2043B3    JSR    $B343    ;扫描键盘，读入用户输入的内容
780B    A000      LDY    #$00
780D    8490      STY    $90      ;若是读内存则：
780F    F026      BEQ    $7837    取首地址存入$8C、$8D
7811    A203      LDX    #$03     取末地址存入$8E、$8F
```

```
7813   0A       ASL              置$90=HA7
7814   0A       ASL              ;若是写内存则:
7815   0A       ASL              取首地址存入$8C、$8D
7816   0A       ASL              取写入数据存入$8E、$8F
7817   0A       ASL              置$90=H93
7818   268E     ROL    $8E
781A   268F     ROL    $8F
781C   CA       DEX
781D   10F8     BPL    $7817
781F   A590     LDA    $90
7821   D004     BNE    $7827
7823   B58F     LDA    $8F,X
7825   958D     STA    $8D,X
7827   E8       INX
7828   F0F5     BEQ    $781F
782A   D011     BNE    $783D
782C   206B78   JSR    $786B
782F   4C0078   JMP    $7800
7832   206B78   JSR    $786B
7835   A425     LDY    $25
    读数判断子程序
7837   A200     LDX    #$00
7839   868E     STX    $8E
783B   868F     STX    $8F
783D   B90105   LDA    $0501,Y
7840   C8       INY
7841   4930     EOR    #$30
7843   C90A     CMP    #$0A
7845   90CA     BCC    $7811
7847   6988     ADC    #$88
7849   C9FA     CMP    #$FA
784B   B0C4     BCS    $7811
784D   8425     STY    $25
784F   C9B9     CMP    #$B9
7851   F0D9     BEQ    $782C
7853   C999     CMP    #$99
7855   F0DB     BEQ    $7832
7857   C9A7     CMP    #$A7
7859   F00B     BEQ    $7866
785B   C993     CMP    #$93
785D   F007     BEQ    $7866
785F   C9EA     CMP    #$EA
7861   D09D     BNE    $7800
7863   4C1D82   JMP    $821D
```

```
7866    8590      STA    $90
7868    4C3578    JMP    $7835
786B    A590      LDA    $90
786D    F026      BEQ    $7895
786F    C9A7      CMP    #$A7
7871    F022      BEQ    $7895
7873    C993      CMP    #$93
7875    D089      BNE    $7800
7877    A58E      LDA    $8E         ;写内存的处理
7879    A000      LDY    #$00
787B    918C      STA    ($8C),Y
787D    4C8878    JMP    $7888
7880    A58C      LDA    $8C
7882    C58E      CMP    $8E
7884    A58D      LDA    $8D
7886    E58F      SBC    $8F
7888    E68C      INC    $8C
788A    D002      BNE    $788E
788C    E68D      INC    $8D
788E    60        RTS
788F    A58C      LDA    $8C
7891    2907      AND    #$07
7893    D00D      BNE    $78A2
7895    2007AC    JSR    $AC07       ;读内存的处理
7898    A58D      LDA    $8D
789A    20B478    JSR    $78B4
789D    A58C      LDA    $8C
789F    20B478    JSR    $78B4
78A2    A920      LDA    #$20
78A4    2034AC    JSR    $AC34
78A7    A000      LDY    #$00
78A9    B18C      LDA    ($8C),Y
78AB    20B478    JSR    $78B4
78AE    208078    JSR    $7880
78B1    90DC      BCC    $788F
78B3    60        RTS
        送显数据子程序
78B4    48        PHA
78B5    4A        LSR
78B6    4A        LSR
78B7    4A        LSR
78B8    4A        LSR
78B9    20BD7F    JSR    $7FBD
78BC    68        PLA
```

```
78BD    290F      AND    #$0F
78BF    0930      ORA    #$30
78C1    C93A      CMP    #$3A
78C3    9002      BCC    $78C7
78C5    6906      ADC    #$06
78C7    4C34AC    JMP    $AC34
```

程序 No. 6－1 由于调用了系统的光标控制、送显、键盘扫描处理子程序，故程序短小精悍，功能较强。显而易见，程序 No. 6－1 只适用于 BS・2A 版本的 F BASIC 系统，如用于其它系统则要改变调用的几个系统子程序入口。

为便于我们以上编写的机器语言程序使用，我们把 No. 6－1 安放在 $7800 开始的一段 RAM 区。若用户的 F BASIC 程序较长时，可把它往后移，或机器语言程序较大时把它向前移。移动时要注意变更几个绝对地址。

读者可把 No. 6－1 的数据编一个 F BASIC 写入程序存入磁带，使用时，把它调入内存，运行一次后删除，即可调用小监控系统。

6・2　6527 反汇编

程序 No. 6－1 只能读出内存的数据，但不便于阅读。程序 No. 6－ 2 可反汇编出任一起始地址的内存程序，其反汇编格式与中华学习机相同。运行程序后直接输入十六进制的反汇编起始地址，回车后即开始反汇编。每反汇编二十行自动暂停，按任一键后继续反汇编。按 STOP 键反汇编结束。

为适用于不同的 F BASIC 版本，No. 6－2 使用 F BASIC 语言编写，反汇编程序清单如下：

```
5 REM 6527 反汇编
10 CLS: INPUT " 输入反汇编首地址 &H";N$ :(注——该句应为英文) N= VAL( "&" + N$ )
20 PRINT : DD=N : GOSUB 2000
30 AA= PEEK (N) : I=0 : RESTORE
40 SS= N : GOSUB 1000 : N= N + 1
50 READ A$ , B$ , C : A= VAL ("&" + A$ ) : IF AA=A THEN 80
60 I=I + 1 : IF I < 151 THEN 50
70 B$ = "???" : C=1
80 IF C=1 THEN POKE 72, 15 : PRINT B$ ; : GOTO 200
90 SS=N : GOSUB 1000 : N=N + 1
100 IF C= 2 THEN 120
110 SS= N : GOSUB 1000 : N= N + 1
120 POKE 72, 15
130 FOR K=1 TO LEN (B$ ) : D$ = MID$ (B$ , K, 1)
140 IF D$ = "3" THEN GOSUB 1100
150 IF D$ = "2" THEN SS= N - 1 : GOSUB 1000 : GOTO 190
160 IF D$ = "1" THEN SS= N - 2 : GOSUB 1000 : GOTO 190
170 IF D$ = "/" THEN D$ = ","
180 PRINT D$ ;
```

```
190 NEXT
200 H= H + 1 : IF H < 20 THEN 20
210 PAUSE : H= 0 : GOTO 20
1000 SS$ = HEX$ ( PEEK( SS ))
1010 IF LEN( SS$ )= 1 THEN SS$ = "0" + SS$
1020 PRINT SS$ ; : RETURN
1100 S= VAL( "&" + SS$ ) : IF S < 128 THEN E= N + S : GOTO 1120
1110 E= N - 256 + S
1120 DD= E : GOSUB 2000
1130 RETURN 190
2000 DD$ = HEX$ ( DD )
2010 IF LEN( DD$ ) < 4 THEN DD$ = "0" + DD$ :GOTO 2010
2020 PRINT DD$ " " ; : RETURN
220 DATA 69,ADC # $2,2,6D,ADC $21,3,65,ADC $2,2,61,ADC ($2/X),2,71,ADC ($2)/Y,2,
    75,ADC $2/X,2,7D,ADC $21/X,3,79,ADC $21/Y,3,29,AND # $2,2,2D,AND $21,3,25,AND
    $2,2,21,AND ($2/X),2
230 DATA 31,AND ($2)/Y,2,35,AND $2/X,2,3D,AND $21/X,3,39,AND $21/Y,3,E,ASL $21,
    3,6,ASL $2,2,A,ASL,1,16,ASL $2/X,2,1E,ASL $21/X,3,90,BCC $3,2,B0,BCS $3,2,F0,
    BEQ $3,2,D0,BNE $3,2,50,BVC $3,2
240 DATA 70,BVS $3,2,30,BMI $3,2,10,BPL $3,2,24,BIT $2,2,2C,BIT $21,3,0,BRK,1,18,
    CLC,1,D8,CLD,1,58,CLI,1,B8,CLV,1,C9,CMP # $2,2,CD,CMP $21,3,C1,CMP ($2/X),2,
    D1,CMP ($2)/Y,2,D5,CMP $2/X,2,
250 DATA DD,CMP $21/X,3,D9,CMP $21/Y,3,C5,CMP $2,2,E0,CPX # $2,2,E4,CPX $2,2,EC,
    CPX $21,3,C0,CPY # $2,2,C4,CPY $2,2,CC,CPY $21,3,C6,DEC $2,2,D6,DEC $2/X,2,CE,
    DEC $21,3,DE,DEC $21/X,3
260 DATA CA,DEX,1,88,DEY,1,49,EOR # $2,2,45,EOR $2,2,55,EOR $2/X,2,4D,EOR $21,3,
    5D,EOR $21/X,3,59,EOR $21/Y,3,41,EOR ($2/X),2,51,EOR ($2)/Y,2,E6,INC $2,2,F6,
    INC $2/X,2,EE,INC $21,3
270 DATA FE,INC $21/X,3,E8,INX,1,C8,INY,1,4C,JMP $21,3,6C,JMP ($21),3,20,JSR $21,3,
    A9,LDA # $2,2,A5,LDA $2,2,B5,LDA $2/X,2,AD,LDA $21,3,BD,LDA $21/X,3,B9,LDA
    $21/Y,3,A1,LDA ($2/X),2,B1,LDA ($2)/Y,2
280 DATA A2,LDX # $2,2,A6,LDX $2,2,B6,LDX $2/Y,2,AE,LDX $21,3,BE,LDX $21/Y,3,A0,
    LDY # $2,2,A4,LDY $2,2,B4,LDY $2/X,2,AC,LDY $21,3,BC,LDY $21/X,3,4A,LSR,1,46,
    LSR $2,2,56,LSR $2/X,2,4E,LSR $21,3
290 DATA 5E,LSR $21/X,3,EA,NOP,1,9,ORA # $2,2,5,ORA $2,2,15,ORA $2/X,2,D,ORA
    $21,3,1D,ORA $21/X,3,19,ORA $21/Y,3,1,ORA ($2/X),2,11,ORA ($2)/Y,2,48,PHA,1,
    8,PHP,1,68,PLA,1,28,PLP,1,2A,ROL,1
300 DATA 26,ROL $2,2,36,ROL $2/X,2,2E,ROL $21,3,3E,ROL $21/X,3,6A,ROR,1,66,ROR
    $2,2,76,ROR $2/X,2,6E,ROR $21,3,7E,ROR $21/X,3,40,RTI,1,60,RTS,1,E9,SBC # $2,2,
    E5,SBC $2,2,F5,SBC $2/X,2
310 DATA ED,SBC $21,3,FD,SBC $21/X,3,F9,SBC $21/Y,3,E1,SBC ($2/X),2,F1,SBC ($2)/Y,
    2,38,SEC,1,F8,SED,1,78,SEI,1,85,STA $2,2,95,STA $2/X,2,8D,STA $21,3,9D,STA $21/
    X,3,99,STA $21/Y,3
320 DATA 81,STA ($2/X),2,91,STA ($2)/Y,2,86,STX $2,2,96,STX $2/Y,2,8E,STX $21,3,84,
```

STY $2,2,94,STY $2/X,2,8C,STY $21,3,AA,TAX,1,A8,TAY,1,BA,TSX,1,8A,TXA,1,9A,TXS,1,98,TYA,1

在输入程序 No. 6-2 的数据语句行时，要注意每一个助记符中的空格。程序 No. 6-2 的 DATA 语句中的数据是按助记符的字母顺序装入的，如欲加快反汇编的运行速度，可调整 DATA 语句中的数据次序，把最常用的指令靠前放置。注意——DATA 中的数据每三个为一组，第一个为指令代码；第二个为汇编指令格式；第三个为指令字节数。

6·3 模拟电子琴

为方便读者编写音乐程序，设计出满意的背景音乐，特提供模拟电子琴程序 No. 6-3。该电子琴有两个八度音域(若需要可随意扩展)、七种和弦伴奏、三种拍节选择、十级音高变化、十五级速度调整，读者还可以随意设定演奏音色。程序采用 F BASIC 语言编写，为顺序结构，通俗易懂，用户界面较好，留给读者较大的发挥余地。读者可任意扩展音域，改变和弦、拍节、音色、音高等，是一个较实用的电子琴模拟程序。

```
5 REM No. 6-3 模拟电子琴
10 CLS : LOCATE 5,5 : PRINT "* * * ANALOG ORGAN * * *" :LOCATE 7,8 : PRINT "Q W E R
   U I O P" :LOCATE 0,9 : PRINT "OCTAVE:1 2 3 4   5 6 7 1"
12 LOCATE 7,11: PRINT "A S D F   J K L ;" :LOCATE 0,12: PRINT "OCTAVE:1 2 3 4   5 6 7 1"
14 LOCATE 4,14: PRINT "TONE(1): 5 6 7 8 9 0" :LOCATE 4,15: PRINT "TONE(2): % & ' ( )"
16 LOCATE 4,16: PRINT "PITCH: , . TEMPO: < >" :LOCATE 4,17: PRINT "CHORD: H CHORD
   SWITCH: Y"
18 LOCATE 4,18: PRINT "NETRONOME: 2 3 4" : LOCATE 6,22: PRINT "WELCOME PLAY......"
20 DIM TN(32), H$(6) : H=0
30 FOR I=1 TO 32 : READ A$ : TN(I)=VAL("&"+A$) : NEXT
40 DATA FD,EE,E1,D4,C8,BD,B2,A8,9F,96,8D,85,7E,76,70,69,63,5E,58,53,4F,4A,46,42,3E,3A,
   37,34,31,2E,2B,29
50 FOR I=0 TO 6 : READ H$(I) : NEXT
60 DATA 00040704,02050905,04071107,00050905,02071107,00040904,05071107
70 A$="" : B$="" : CH=3 : T=15 : DE=1 : LE=4 : D=0 :PAI=4 : CT=-1 : M1=&4F : M2=
   11 : C=8
80 POKE &4000,M1,248,0,0,M2,248,0,0,7,248
90 POKE &4015,15
100 FOR I=1 TO 2 STEP 0 :A$=INKEY$
110 IF A$=" A " THEN D=TN(LE)
120 IF A$=" S " THEN D=TN(LE+2)
130 IF A$=" D " THEN D=TN(LE+4)
140 IF A$=" F " THEN D=TN(LE+5)
150 IF A$=" J " THEN D=TN(LE+7)
160 IF A$=" K " THEN D=TN(LE+9)
170 IF A$=" L " THEN D=TN(LE+11)
180 IF A$=" ; " THEN D=TN(LE+12)
190 IF A$=" Q " THEN D=TN(LE+12)
200 IF A$=" W " THEN D=TN(LE+14)
```

```
210 IF A$="E" THEN D=TN(LE+16)
220 IF A$="R" THEN D=TN(LE+17)
230 IF A$="U" THEN D=TN(LE+19)
240 IF A$="I" THEN D=TN(LE+21)
250 IF A$="O" THEN D=TN(LE+23)
260 IF A$="P" THEN D=TN(LE+24)
270 IF A$="H" THEN H=H+1 : IF H>7 THEN H=0
280 IF A$="," THEN LE=LE-1 : IF LE=0 THEN LE=8
290 IF A$="." THEN LE=LE+1 : IF LE=9 THEN LE=0
300 IF A$="5" THEN M1=&0F
310 IF A$="6" THEN M1=&4F
320 IF A$="7" THEN M1=&8F
330 IF A$="8" THEN M1=&3F
340 IF A$="9" THEN M1=&7F
350 IF A$="0" THEN M1=&BF
360 IF A$="%" THEN M2=&0B
370 IF A$="'" THEN M2=&8B
380 IF A$="(" THEN M2=&37
390 IF A$=")" THEN M2=&77
400 IF A$="&" THEN M2=&4B
410 IF A$="Y" AND CH=3 THEN CH=7 : GOTO 430
420 IF A$="Y" AND CH=7 THEN CH=3
430 IF A$="<" THEN T=T-1 : IF T=0 THEN T=1
440 IF A$=">" THEN T=T+1 : IF T=16 THEN T=15
450 GOSUB 600
460 IF B$=A$ THEN NEXT
470 B$=A$
480 IF A$="" THEN NEXT
490 POKE &4000,M1,248,D,C,M2,248,D+DE,C
500 NEXT
510 END
600 CT=CT+1 : IF CT=256 THEN CT=0
610 IF CT MOD T <> 0 THEN RETURN
620 DI=( CT / T ) MOD PAI
630 ON DI GOTO 660, 700, 740
640 POKE &4008,248,248,TN(VAL(MID$(H$(H),1,2)))-DE,C
650 POKE &4015,CH : RETURN
660 POKE &4008,134,248,TN(VAL(MID$(H$(H),3,2)))-DE,C
670 POKE &4015,CH
680 IF PAI=2 THEN 760
690 RETURN
700 POKE &400A,TN(VAL(MID$(H$(H),5,2)))-DE,C
710 POKE &4015,CH
720 IF PAI=2 THEN 760
```

```
730 RETURN
740 POKE &400A,TN(VAL(MID$(H$(H),7,2)))-DE,C
750 POKE &4015,CH
760 CT=-T
770 RETURN
```

英语单词说明：

OCTAVE：八度音　　TONE：音色

PITCH：音调　　TEMPO：速度

CHORD：和弦　　SWITCH：开关

NETRONOME：节拍

变量说明：

H$——和弦数组。H$(0)～H$(6)分别对应七种和弦。

CH——和弦开关

T——速度

PAI——节拍

DE——音色设定

LE——基准音调

第七章　BS.2A 版本 F BASIC 系统程序及说明

BS.2A 版本的 F BASIC 系统程序共占 32K 字节，地址分配在＄8000～＄FFFF 空间。由于该软件采用交叉结构编写，同一项功能的处理程序被分成几段，安排在其它功能的处理程序中，故很难归类划段。编者试验了多种分类方法都有缺陷。如按程序功能分段，虽然同一功能程序集中了，但程序的地址顺序打乱了；如按程序的地址排序分段则功能相同的程序被分成了几段，段号分散……。最后决定采用不加段号多写注释的方法编写如下。

```
8000  4CAD80    JMP   $80AD   ;执行进入 F BASIC 工作系统的操作
8003  4C88AF    JMP   $AF88   ;转读键处理
```

发声音调数据库

系统发声中的音调数据，每两个数据一组，前一个为音调细调，后一个为音调粗调。为便于应用，特用简谱标注。标注中低一个八度音用一个英文字母“D”作前缀；高一个八度音则用“G”作前缀；中音无前缀。因此，“DD”表示低两个八度；“GGG”表示高三个八度。

8006				AE 0E	4E 0E	F3 0D	9E 0D	4D 0D
				DD 1	DD1#	DD 2	DD 2#	DD 3
8010	01 0D	B9 0C	75 0C	35 0C	F8 0B	BF 0B	89 0B	57 0B
	DD3#4	DD4#	DD5	DD5#	DD6	DD6#	DD7	DD7#D1
8020	27 0B	F9 0A	CF 0A	A6 0A	80 0A	5C 0A	3A 0A	1A 0A
	D1#	D2	D2#	D3	D3# 4	D4#	D5	D5#
8030	FC 09	DF 09	C4 09	AB 09	93 09	7C 09	67 09	52 09
	D6	D6#	D7	D7#1	1#	2	2#	3
8040	3F 09	2D 09	1C 09	0C 09	FD 08	EE 08	E1 08	D4 08
	3# 4	4#	5	5#	6	6#	7	7# G1
8050	C8 08	BD 08	B2 08	A8 08	9F 08	96 08	8D 08	85 08
	G1#	G2	G2#	G3	G3#G4	G4#	G5	G5#
8060	7E 08	76 08	70 08	69 08	63 08	5E 08	58 08	53 08
	G6	G6#	G7	G7#GG1	GG1#	GG2	GG2#	GG3
8070	4F 08	4A 08	46 08	42 08	3E 08	3A 08	37 08	34 08
	GG3#	GG4#	GG5	GG5#	GG6	GG6#	GG7	GG7#GGG1
8080	31 08	2E 08	2B 08	29 08	27 08	24 08	22 08	20 08
	GGG1#	GGG2	GGG2#	GGG3	GGG4	GGG4#	GGG5	GGG5#
8090	1E 08	1C 08	1B 08					
	GGG6	GGG6#	GGG7					

PLAY 语句的执行参数

```
8096                    01 02 03 04 06 08 0C 10 18 20
80A0  12 16 00 04 08 0A 0E 00 18 30 48 60 78
                           O0O1O2O3O4O5
```

进入 F BASIC 工作系统的操作

```
;置中断转移入口为$880F
80AD    A94C      LDA    #$4C
80AF    85ED      STA    $ED
80B1    A90F      LDA    #$0F
80B3    85EE      STA    $EE
80B5    A988      LDA    #$88
80B7    85EF      STA    $EF
;对有关单元置初值
80B9    2082AA    JSR    $AA82
;绘制F BASIC选项画面
80BC    203181    JSR    $8131
;定义光标
80BF    20A9AD    JSR    $ADA9
;读按键
80C2    2088AF    JSR    $AF88
80C5    C931      CMP    #$31
;按"1"键转$80F0
80C7    F027      BEQ    $80F0
80C9    C932      CMP    #$32
;按"2"键转$80D4
80CB    F007      BEQ    $80D4
80CD    C933      CMP    #$33
;不是"1、2、3"键则转$80BF
80CF    D0EE      BNE    $80BF
;按"3"键则返回复位画面
80D1    4C00C4    JMP    $C400
;转BG GRAPHIC绘图
80D4    4CAFB5    JMP    $B5AF
    SYSTEM保留字入口
80D7    A95A      LDA    #$5A
80D9    8D3A70    STA    $703A
80DC    A933      LDA    #$33
80DE    8D3B70    STA    $703B
80E1    8500      STA    $00
80E3    A507      LDA    $07
80E5    8D3C70    STA    $703C
80E8    A508      LDA    $08
80EA    8D3D70    STA    $703D
80ED    4CAD80    JMP    $80AD
    从选项画面进入BASIC
80F0    2088AF    JSR    $AF88
80F3    D0FB      BNE    $80F0
80F5    A500      LDA    $00
80F7    C959      CMP    #$59
80F9    F026      BEQ    $8121
80FB    AD3A70    LDA    $703A
80FE    C95A      CMP    #$5A
8100    D025      BNE    $8127
8102    AD3B70    LDA    $703B
8105    C933      CMP    #$33
8107    D01E      BNE    $8127
8109    AD3C70    LDA    $703C
810C    8507      STA    $07
810E    AD3D70    LDA    $703D
8111    8508      STA    $08
8113    A93E      LDA    #$3E
;置F BASIC程序存放首址于$05、$06单元
8115    8505      STA    $05
8117    A970      LDA    #$70
8119    8506      STA    $06
;CLEAR处理
811B    202892    JSR    $9228
;CLS处理
811E    204B81    JSR    $814B
;零页有关单元赋值;清屏;送配色号码等
8121    202E85    JSR    $852E
8124    4C0B82    JMP    $820B
;第一次进入F BASIC时,清背景一页
8127    204B81    JSR    $814B
812A    A959      LDA    #$59
812C    8500      STA    $00
812E    4CFF81    JMP    $81FF
    绘制F BASIC选项画面
8131    2088AF    JSR    $AF88
8134    D0FB      BNE    $8131
;画选项画面中的长方框
8136    209C81    JSR    $819C
8139    A900      LDA    #$00
;送显选项画面中的内容
F BASIC
1--BASIC
2--BG GRAPHIC
3--END
1,2,3 KEY IN !!
813B    8544      STA    $44
813D    A544      LDA    $44
813F    C905      CMP    #$05
8141    F007      BEQ    $814A
```

```
8143    205481    JSR    $8154
8146    E644      INC    $44
8148    D0F3      BNE    $813D
814A    60        RTS
814B    A59E      LDA    $9E
814D    C981      CMP    #$81
814F    F0F9      BEQ    $814A
;清屏,清背景一页
8151    4C71AB    JMP    $AB71
;显示选项画面中的字符提示
8154    0A        ASL
8155    0A        ASL
8156    AA        TAX
8157    BD91C3    LDA    $C391,X
815A    8548      STA    $48
815C    BD92C3    LDA    $C392,X
;确定显示起点座标
815F    8549      STA    $49
8161    BD93C3    LDA    $C393,X
8164    8552      STA    $52
8166    BD94C3    LDA    $C394,X
;确定取数地址首址
8169    8553      STA    $53
816B    4CF887    JMP    $87F8
```

F BASIC 选项画面的数据

```
816E                          31 2D
                              1  —
8170  2D 42 41 53 49 43 00 32 2D 2D 42 47 20 47 52 41
      —  B  A  S  I  C     2  —  —  B  G     G  R  A
8180  50 48 49 43 00 33 2D 2D 45 4E 44 00 31 2C 32 2C
      P  H  I  C     3  —  —  E  N  D     1  ,  2  ,
8190  33 20 4B 45 59 20 49 4E 20 21 21 00
      3     K  E  Y     I  N     !  !
```

```
;绘制 F BASIC 选项画面的长方框
819C    20F1AB    JSR    $ABF1
819F    A9DE      LDA    #$DE
81A1    85AD      STA    $AD
81A3    A920      LDA    #$20
81A5    A2C5      LDX    #$C5
;置 PPU 首址 $20C5
81A7    20D381    JSR    $81D3
81AA    A9E0      LDA    #$E0
81AC    85AD      STA    $AD
81AE    A922      LDA    #$22
81B0    A2E5      LDX    #$E5
;置 PPU 首址 $22E5
81B2    20D381    JSR    $81D3
81B5    A532      LDA    $32
81B7    0904      ORA    #$04
81B9    8D0020    STA    $2000
81BC    A920      LDA    #$20
81BE    A2E5      LDX    #$E5
;置 PPU 首址 $20E5
81C0    20EA81    JSR    $81EA
81C3    A920      LDA    #$20
81C5    A2FA      LDX    #$FA
;置 PPU 首址 $20FA
81C7    20EA81    JSR    $81EA
81CA    20E6AB    JSR    $ABE6
81CD    20F1B3    JSR    $B3F1
81D0    4CFAAB    JMP    $ABFA
81D3    20F881    JSR    $81F8
81D6    A5AD      LDA    $AD
81D8    8D0720    STA    $2007
81DB    A9E3      LDA    #$E3
81DD    A014      LDY    #$14
81DF    20F181    JSR    $81F1
81E2    E6AD      INC    $AD
81E4    A5AD      LDA    $AD
81E6    8D0720    STA    $2007
81E9    60        RTS
81EA    20F881    JSR    $81F8
81ED    A9E2      LDA    #$E2
81EF    A010      LDY    #$10
;向 PPU 写入 H10 个 H2E
81F1    8D0720    STA    $2007
81F4    88        DEY
81F5    D0FA      BNE    $81F1
81F7    60        RTS
```

```
81F8    8D0620    STA    $2006
81FB    8E0620    STX    $2006
81FE    60        RTS
;第一次进入 F BASIC 时置初值
81FF    A2FF      LDX    #$FF
;设栈顶为 HFF
8201    9A        TXS
8202    202E85    JSR    $852E
8205    208883    JSR    $8388
;送显"F BASIC BS. 2ACNEDC SHEN
ZHEN CO. END 保留字入口
8208    2009AA    JSR    $AA09
;令 $24=0
820B    203B87    JSR    $873B
820E    8D3A70    STA    $703A
8211    8D3B70    STA    $703B
8214    A2FF      LDX    #$FF
;设置堆栈顶为 HFF
8216    9A        TXS
8217    8A        TXA
8218    48        PHA
;栈底存入两个 HFF
8219    48        PHA
821A    20A9AD    JSR    $ADA9
821D    BA        TSX
821E    8617      STX    $17
8220    58        CLI
8221    A617      LDX    $17
8223    9A        TXS
;令光标位于行首
8224    2003AC    JSR    $AC03
;送显"OK"
8227    20FFA9    JSR    $A9FF
;若 $24=0,使 CONT 指令失效
822A    A524      LDA    $24
822C    F005      BEQ    $8233
;送显"."，若"OK"后有点则 CONT 有效；
若"OK"后无点则 CONT 无效
822E    A92E      LDA    #$2E
8230    200DAC    JSR    $AC0D
8233    2003AC    JSR    $AC03
        键盘输入的 F BASIC 程序的翻译
8236    2043B3    JSR    $B343
8239    20328A    JSR    $8A32
823C    20C78C    JSR    $8CC7
823F    C903      CMP    #$03
8241    F0F3      BEQ    $8236
8243    20D48C    JSR    $8CD4
8246    29FF      AND    #$FF
8248    F0EC      BEQ    $8236
;若 A=H30～H39 则令 C=0；否则令 C=1
824A    20DD8C    JSR    $8CDD
824D    B010      BCS    $825F
        暂缓执行型程序的翻译
;行号的翻译
824F    20B183    JSR    $83B1
;进行 CLEAR 处理
8252    202892    JSR    $9228
8255    4C3682    JMP    $8236
8258    A9FF      LDA    #$FF
825A    850B      STA    $0B
825C    850C      STA    $0C
825E    60        RTS
        立即执行型程序的翻译与执行
825F    20598E    JSR    $8E59
8262    A944      LDA    #$44
8264    850D      STA    $0D
8266    A9B1      LDA    #$B1
8268    850E      STA    $0E
826A    A900      LDA    #$00
826C    8509      STA    $09
826E    A903      LDA    #$03
8270    850A      STA    $0A
8272    205882    JSR    $8258
;扫描 STOP 键
8275    20D486    JSR    $86D4
;检查堆栈空间
8278    205B8A    JSR    $8A5B
;读数
827B    20D48C    JSR    $8CD4
827E    48        PHA
827F    A509      LDA    $09
8281    8558      STA    $58
8283    A50A      LDA    $0A
8285    8559      STA    $59
8287    20A18C    JSR    $8CA1
828A    68        PLA
828B    F04D      BEQ    $82DA
```

```
828D    C93A      CMP    #$3A
;是":"转$8275
828F    F0E4      BEQ    $8275
8291    C927      CMP    #$27
;是"'"转$8345
8293    F045      BEQ    $82DA
8295    C980      CMP    #$80
;是保留字转$82DD
8297    B044      BCS    $82DD
;检查是否是大写字母
8299    20C3A1    JSR    $A1C3
;不是转$82D7
829C    B039      BCS    $82D7
829E    48        PHA
829F    20BB8C    JSR    $8CBB
82A2    68        PLA
;变量的判断处理
82A3    20759F    JSR    $9F75
82A6    A590      LDA    $90
82A8    48        PHA
82A9    A58C      LDA    $8C
82AB    48        PHA
82AC    A58D      LDA    $8D
82AE    48        PHA
82AF    20A285    JSR    $85A2
;取数判断嵌套处理
82B2    20829C    JSR    $9C82
82B5    68        PLA
82B6    858D      STA    $8D
82B8    68        PLA
82B9    858C      STA    $8C
82BB    68        PLA
82BC    8590      STA    $90
82BE    A000      LDY    #$00
82C0    290F      AND    #$0F
82C2    C903      CMP    #$03
82C4    F01D      BEQ    $82E3
82C6    20EFA3    JSR    $A3EF
82C9    AD0004    LDA    $0400
82CC    918C      STA    ($8C),Y
82CE    AD0104    LDA    $0401
82D1    C8        INY
82D2    918C      STA    ($8C),Y
82D4    4C7582    JMP    $8275
;出错处理,代码=1
82D7    4C6D83    JMP    $836D
;转非处理语句的处理
82DA    4C4583    JMP    $8345
;是保留字的处理
82DD    4C2183    JMP    $8321
;出错处理,代码=4
82E0    4C7083    JMP    $8370
82E3    20F2A3    JSR    $A3F2
;串变量的处理
82E6    20EF82    JSR    $82EF
82E9    4C7582    JMP    $8275
82EC    918C      STA    ($8C),Y
82EE    60        RTS
```

字符串变量赋值。把由$400指出的串长，$401、$402指出字串首址的串值取出送入变量区

```
82EF    A000      LDY    #$00
82F1    AD0004    LDA    $0400
82F4    F0F6      BEQ    $82EC
82F6    C920      CMP    #$20
;当串长大于H20时则出错
82F8    B0E6      BCS    $82E0
;向变量区送入串长
82FA    918C      STA    ($8C),Y
82FC    AD0104    LDA    $0401
82FF    8519      STA    $19
8301    AD0204    LDA    $0402
8304    851A      STA    $1A
8306    C8        INY
8307    98        TYA
8308    18        CLC
8309    658C      ADC    $8C
830B    858C      STA    $8C
830D    A900      LDA    #$00
830F    658D      ADC    $8D
;调整变量区地址指针
8311    858D      STA    $8D
8313    AE0004    LDX    $0400
8316    A000      LDY    #$00
8318    B119      LDA    ($19),Y
831A    918C      STA    ($8C),Y
831C    C8        INY
831D    CA        DEX
```

```
;把串值存入变量区
831E    D0F8      BNE    $8318
8320    60        RTS
    保留字的转向处理
8321    38        SEC
8322    E980      SBC    #$80
8324    C944      CMP    #$44
8326    B045      BCS    $836D
8328    0A        ASL
8329    AA        TAX
832A    BDC6C2    LDA    $C2C6,X
832D    8519      STA    $19
832F    BDC7C2    LDA    $C2C7,X
8332    851A      STA    $1A
8334    BA        TSX
8335    8617      STX    $17
8337    207883    JSR    $8378
833A    204B9D    JSR    $9D4B
833D    4C7582    JMP    $8275
;进入 END 的后续处理
8340    4C1D82    JMP    $821D
    REM、DATA 非执行语句的处理
8343    68        PLA
8344    68        PLA
8345    A50D      LDA    $0D
8347    8509      STA    $09
8349    A50E      LDA    $0E
;取下一行程序首址
834B    850A      STA    $0A
834D    20C78C    JSR    $8CC7
8350    F0EE      BEQ    $8340
8352    18        CLC
8353    6509      ADC    $09
8355    850D      STA    $0D
8357    A900      LDA    #$00
8359    650A      ADC    $0A
;存下一行程序末址
835B    850E      STA    $0E
835D    C8        INY
835E    B109      LDA    ($09),Y
8360    850B      STA    $0B
8362    C8        INY
8363    B109      LDA    ($09),Y
;取下一程序行行号
8365    850C      STA    $0C
;令$09、$0A 指向程序体
8367    202799    JSR    $9927
836A    4C7582    JMP    $8275
    THEN、TO、STEP、OFF、
    SPC、TAB 保留字的处理
;A=1,错误类型为"SN"
836D    A901      LDA    #$01
;A=4,错误类型为"IL"
836F    AEA904    LDX    $04A9
8372    4C8FA9    JMP    $A98F
;读数处理
8375    20A18C    JSR    $8CA1
8378    20D48C    JSR    $8CD4
837B    29FF      AND    #$FF
837D    F002      BEQ    $8381
837F    C93A      CMP    #$3A
8381    60        RTS
    NEW 保留字入口
8382    208883    JSR    $8388
8385    4C1482    JMP    $8214
;令$24=0
8388    203B87    JSR    $873B
;令$703E=$703F=$7040=0
程序存储区复位
838B    8D3E70    STA    $703E
838E    8D3F70    STA    $703F
8391    8D4070    STA    $7040
8394    A23E      LDX    #$3E
;令$05=H3E
8396    8605      STX    $05
8398    E8        INX
;令$07=H3F
8399    8607      STX    $07
839B    E8        INX
;令$1D=H40
839C    861D      STX    $1D
839E    E8        INX
;令$1F=H41
839F    861F      STX    $1F
83A1    E8        INX
;令$21=H42
83A2    8621      STX    $21
```

```
;以上置数使：
程序区首址＝$703E；
程序区末址＝$703F；
变量区首址＝$7040；
变量区末址＝$7041；
自由 RAM 区首址＝$7042
83A4    A970      LDA    #$70
83A6    8506      STA    $06
83A8    8508      STA    $08
83AA    851E      STA    $1E
83AC    8520      STA    $20
83AE    8522      STA    $22
83B0    60        RTS
        翻译后的程序存入程序区的存前预处理
;令$24=0
83B1    203B87    JSR    $873B
;令$0B=$0C=HFF
83B4    205882    JSR    $8258
83B7    20098D    JSR    $8D09
83BA    A9FF      LDA    #$FF
83BC    C528      CMP    $28
83BE    D006      BNE    $83C6
83C0    C529      CMP    $29
83C2    D002      BNE    $83C6
83C4    F0AA      BEQ    $8370
83C6    20C78C    JSR    $8CC7
83C9    C920      CMP    #$20
83CB    D003      BNE    $83D0
83CD    20AD8C    JSR    $8CAD
83D0    29FF      AND    #$FF
;若行号后一个单元的内容为零则删除该行
83D2    F02D      BEQ    $8401
83D4    A528      LDA    $28
83D6    8544      STA    $44
83D8    A529      LDA    $29
83DA    8545      STA    $45
;翻译判断处理
83DC    20598E    JSR    $8E59
83DF    38        SEC
83E0    A54D      LDA    $4D
83E2    E900      SBC    #$00
83E4    8528      STA    $28
83E6    A54E      LDA    $4E
83E8    E903      SBC    #$03
83EA    8529      STA    $29
83EC    A902      LDA    #$02
83EE    18        CLC
83EF    6528      ADC    $28
83F1    855A      STA    $5A
83F3    A544      LDA    $44
83F5    850F      STA    $0F
83F7    A545      LDA    $45
83F9    8510      STA    $10
;检查程序区是否存有程序
83FB    204884    JSR    $8448
83FE    4C8A84    JMP    $848A
8401    4C4084    JMP    $8440
8404    A505      LDA    $05
8406    855B      STA    $5B
8408    A506      LDA    $06
;程序区首址存入$5B、$5C
840A    855C      STA    $5C
840C    A000      LDY    #$00
;取本行程序占用内存单元数
840E    B15B      LDA    ($5B),Y
8410    D003      BNE    $8415
8412    A9FF      LDA    #$FF
8414    60        RTS
        非空程序区的程序存入
8415    C8        INY
8416    B15B      LDA    ($5B),Y
8418    852C      STA    $2C
841A    C8        INY
841B    B15B      LDA    ($5B),Y
;取行号存入$2C、$2D
841D    852D      STA    $2D
841F    A544      LDA    $44
8421    8528      STA    $28
8423    A545      LDA    $45
8425    8529      STA    $29
8427    20F78B    JSR    $8BF7
842A    F0E8      BEQ    $8414
842C    90E6      BCC    $8414
842E    A000      LDY    #$00
8430    18        CLC
8431    B15B      LDA    ($5B),Y
8433    655B      ADC    $5B
8435    855B      STA    $5B
```

```
8437    A55C      LDA    $5C
8439    6900      ADC    #$00
843B    855C      STA    $5C
843D    4C0C84    JMP    $840C
  删除程序行
8440    A528      LDA    $28
8442    8544      STA    $44
8444    A529      LDA    $29
8446    8545      STA    $45
;判断程序区是否存有程序
8448    200484    JSR    $8404
844B    D0C7      BNE    $8414
844D    A000      LDY    #$00
844F    B15B      LDA    ($5B),Y
8451    853C      STA    $3C
8453    18        CLC
8454    655B      ADC    $5B
8456    850B      STA    $0B
8458    A55C      LDA    $5C
845A    6900      ADC    #$00
845C    850C      STA    $0C
845E    A000      LDY    #$00
8460    B10B      LDA    ($0B),Y
8462    915B      STA    ($5B),Y
8464    E60B      INC    $0B
8466    D002      BNE    $846A
8468    E60C      INC    $0C
846A    E65B      INC    $5B
846C    D002      BNE    $8470
846E    E65C      INC    $5C
8470    A55C      LDA    $5C
8472    C508      CMP    $08
8474    D0E8      BNE    $845E
8476    A55B      LDA    $5B
8478    C507      CMP    $07
847A    D0E2      BNE    $845E
847C    38        SEC
847D    A507      LDA    $07
847F    E53C      SBC    $3C
8481    8507      STA    $07
8483    A508      LDA    $08
8485    E900      SBC    #$00
8487    8508      STA    $08
8489    60        RTS
  空程序区的程序存入
848A    A507      LDA    $07
848C    18        CLC
848D    655A      ADC    $5A
848F    8528      STA    $28
8491    A508      LDA    $08
8493    6900      ADC    #$00
8495    8529      STA    $29
;检查 RAM 自由区,有富余则返回
8497    20448A    JSR    $8A44
849A    200484    JSR    $8404
849D    A507      LDA    $07
849F    850B      STA    $0B
84A1    18        CLC
84A2    655A      ADC    $5A
84A4    8519      STA    $19
84A6    A508      LDA    $08
84A8    850C      STA    $0C
84AA    6900      ADC    #$00
84AC    851A      STA    $1A
84AE    A000      LDY    #$00
84B0    B10B      LDA    ($0B),Y
84B2    9119      STA    ($19),Y
84B4    201685    JSR    $8516
84B7    202285    JSR    $8522
84BA    A55B      LDA    $5B
84BC    C50B      CMP    $0B
84BE    D0EE      BNE    $84AE
84C0    A55C      LDA    $5C
84C2    C50C      CMP    $0C
84C4    D0E8      BNE    $84AE
84C6    B10B      LDA    ($0B),Y
84C8    9119      STA    ($19),Y
84CA    A000      LDY    #$00
84CC    A55A      LDA    $5A
;存入程序行长度
84CE    915B      STA    ($5B),Y
84D0    C8        INY
84D1    A50F      LDA    $0F
;存入行号低位
84D3    915B      STA    ($5B),Y
84D5    C8        INY
84D6    A510      LDA    $10
;存入行号高位
```

```
84D8    915B    STA     ($5B),Y
84DA    C8      INY
84DB    18      CLC
84DC    A903    LDA     #$03
84DE    655B    ADC     $5B
84E0    855B    STA     $5B
84E2    A900    LDA     #$00
84E4    655C    ADC     $5C
;调整程序区地址指针
84E6    855C    STA     $5C
84E8    A55A    LDA     $5A
84EA    38      SEC
84EB    E903    SBC     #$03
84ED    AA      TAX
84EE    A000    LDY     #$00
84F0    A900    LDA     #$00
84F2    854D    STA     $4D
84F4    A903    LDA     #$03
84F6    854E    STA     $4E
84F8    B14D    LDA     ($4D),Y
84FA    915B    STA     ($5B),Y
84FC    C8      INY
84FD    CA      DEX
;把三页的内容依次送入程序区
84FE    D0F8    BNE     $84F8
8500    18      CLC
8501    A507    LDA     $07
8503    655A    ADC     $5A
8505    8507    STA     $07
8507    A900    LDA     #$00
8509    6508    ADC     $08
;计算新的程序区末址
850B    8508    STA     $08
850D    A000    LDY     #$00
850F    98      TYA
8510    9107    STA     ($07),Y
8512    C8      INY
;最后存入两个零作为程序区结束标志
8513    9107    STA     ($07),Y
8515    60      RTS
8516    A50B    LDA     $0B
8518    F003    BEQ     $851D
851A    C60B    DEC     $0B
851C    60      RTS
851D    C60B    DEC     $0B
851F    C60C    DEC     $0C
8521    60      RTS
8522    A519    LDA     $19
8524    F003    BEQ     $8529
8526    C619    DEC     $19
8528    60      RTS
8529    C619    DEC     $19
852B    C61A    DEC     $1A
852D    60      RTS
        首次进入 F BASIC 的赋值
852E    A904    LDA     #$04
8530    85B9    STA     $B9
8532    85BA    STA     $BA
8534    85BB    STA     $BB
8536    A908    LDA     #$08
8538    85C3    STA     $C3
853A    85C4    STA     $C4
853C    85C5    STA     $C5
853E    A903    LDA     #$03
8540    85BF    STA     $BF
8542    85C0    STA     $C0
8544    85C1    STA     $C1
8546    A90F    LDA     #$0F
8548    85BC    STA     $BC
854A    85BD    STA     $BD
854C    85BE    STA     $BE
854E    A930    LDA     #$30
8550    85CF    STA     $CF
8552    85D0    STA     $D0
8554    85D1    STA     $D1
8556    A902    LDA     #$02
8558    85D2    STA     $D2
855A    85D3    STA     $D3
855C    85D4    STA     $D4
855E    203B87  JSR     $873B
8561    854C    STA     $4C
8563    8557    STA     $57
8565    8597    STA     $97
8567    A9FF    LDA     #$FF
8569    8503    STA     $03
856B    A903    LDA     #$03
856D    8556    STA     $56
856F    A97F    LDA     #$7F
```

```
8571    8504      STA    $04
;令$200~$2FF=HEF
8573    20DFA6    JSR    $A6DF
;对六页赋值
8576    206EA9    JSR    $A96E
8579    4C82AA    JMP    $AA82
```

暂缓执行型程序的运行(RUN 保留字入口)

```
;CLEAR 处理
857C    202892    JSR    $9228
;令$24=0
857F    203B87    JSR    $873B
;对六页有关单元置初值
8582    206EA9    JSR    $A96E
;对二页置入 HEF,清卡通定义区
8585    20DFA6    JSR    $A6DF
;读数
8588    207883    JSR    $8378
858B    D012      BNE    $859F
858D    A200      LDX    #$00
858F    9A        TXS
8590    A9FF      LDA    #$FF
8592    48        PHA
8593    48        PHA
8594    A505      LDA    $05
8596    8509      STA    $09
8598    A506      LDA    $06
859A    850A      STA    $0A
859C    4C4D83    JMP    $834D
859F    4C4691    JMP    $9146
85A2    20D48C    JSR    $8CD4
85A5    C9F6      CMP    #$F6
85A7    D003      BNE    $85AC
85A9    4CA18C    JMP    $8CA1
;出错处理,代码=1
85AC    4C6D83    JMP    $836D
```

内存程序的列表操作

列表处理(LIST 保留字入口)

```
85AF    08        PHP
85B0    A200      LDX    #$00
85B2    8686      STX    $86
;列表首行号置 H0000
85B4    8687      STX    $87
85B6    CA        DEX
85B7    8688      STX    $88
;列表终行号置 HFFFF
85B9    8689      STX    $89
85BB    28        PLP
;若 A=0,进行 LIST 处理
85BC    F046      BEQ    $8604
85BE    C92C      CMP    #$2C
85C0    F023      BEQ    $85E5
85C2    C92D      CMP    #$2D
;若为"LIST,"或"LIST-"则转$85E5
85C4    F01F      BEQ    $85E5
85C6    C90B      CMP    #$0B
;若"LIST"后不是"B"则出错
85C8    D0E2      BNE    $85AC
85CA    20AD8C    JSR    $8CAD
85CD    8586      STA    $86
85CF    8588      STA    $88
85D1    20AD8C    JSR    $8CAD
85D4    8587      STA    $87
85D6    8589      STA    $89
85D8    207583    JSR    $8375
;是"LIST n:"转$8604
85DB    F027      BEQ    $8604
85DD    C92C      CMP    #$2C
;是"LIST n,"转$85E5
85DF    F004      BEQ    $85E5
85E1    C92D      CMP    #$2D
85E3    D0C7      BNE    $85AC
;"LIST n,"或"LIST n-"的处理
85E5    207583    JSR    $8375
85E8    F014      BEQ    $85FE
85EA    C90B      CMP    #$0B
85EC    D0BE      BNE    $85AC
85EE    20AD8C    JSR    $8CAD
85F1    8588      STA    $88
85F3    20AD8C    JSR    $8CAD
85F6    8589      STA    $89
85F8    20A18C    JSR    $8CA1
85FB    4C0486    JMP    $8604
;"LIST n,:"的处理
85FE    A9FF      LDA    #$FF
8600    8588      STA    $88
8602    8589      STA    $89
8604    20F88F    JSR    $8FF8
8607    A586      LDA    $86
```

```
8609   8544     STA   $44
860B   A587     LDA   $87
860D   8545     STA   $45
860F   200484   JSR   $8404
8612   A000     LDY   #$00
8614   B15B     LDA   ($5B),Y
8616   F044     BEQ   $865C
8618   A55B     LDA   $5B
861A   8509     STA   $09
861C   A55C     LDA   $5C
861E   850A     STA   $0A
8620   A000     LDY   #$00
8622   B109     LDA   ($09),Y
8624   F036     BEQ   $865C
8626   C8       INY
```

;取出程序区行号存入$28、$29

取出列表结束行号存入$2C、$2D

```
8627   B109     LDA   ($09),Y
8629   8528     STA   $28
862B   C8       INY
862C   B109     LDA   ($09),Y
862E   8529     STA   $29
8630   A588     LDA   $88
8632   852C     STA   $2C
8634   A589     LDA   $89
8636   852D     STA   $2D
```

;检查是否达到结束行号:

若是结束行号则转$863F

若大于结束行号则转$865C

```
8638   20F78B   JSR   $8BF7
863B   F002     BEQ   $863F
863D   B01D     BCS   $865C
```

;若小于结束行号则顺序执行,读出本行语句反翻译后送往七页,扫描STOP键若没按键则继续列表;若按键不是Esc键则继续列表

```
863F   206D86   JSR   $866D
8642   205686   JSR   $8656
8645   F0D9     BEQ   $8620
8647   C91B     CMP   #$1B
8649   D0D5     BNE   $8620
```

;扫描键盘,若继续按Esc键则停止列表

```
864B   205686   JSR   $8656
864E   F0FB     BEQ   $864B
8650   C91B     CMP   #$1B
8652   F0F7     BEQ   $864B
```

;若不是Esc键则继续列表

```
8654   D0CA     BNE   $8620
```

;扫描STOP

```
8656   20D486   JSR   $86D4
```

;扫描键盘

```
8659   4C88AF   JMP   $AF88
865C   4C0190   JMP   $9001
865F   A93D     LDA   #$3D
8661   18       CLC
8662   6546     ADC   $46
8664   8552     STA   $52
8666   A900     LDA   #$00
8668   8553     STA   $53
866A   4CB787   JMP   $87B7
```

内存程序代码的翻译与显示

```
866D   A900     LDA   #$00
866F   8562     STA   $62
8671   A001     LDY   #$01
8673   B109     LDA   ($09),Y
8675   8528     STA   $28
8677   C8       INY
8678   B109     LDA   ($09),Y
```

;取出程序区的行号转换成十进制的ASCII码送入七页显示

```
867A   8529     STA   $29
867C   203C8D   JSR   $8D3C
867F   20EB8C   JSR   $8CEB
8682   205F86   JSR   $865F
8685   A920     LDA   #$20
8687   20F087   JSR   $87F0
868A   A903     LDA   #$03
868C   18       CLC
868D   6509     ADC   $09
868F   854D     STA   $4D
8691   A900     LDA   #$00
8693   650A     ADC   $0A
8695   854E     STA   $4E
8697   200A90   JSR   $900A
869A   203B8A   JSR   $8A3B
869D   20F887   JSR   $87F8
86A0   A54D     LDA   $4D
86A2   8509     STA   $09
86A4   A54E     LDA   $4E
```

```
86A6    850A      STA    $0A
;令光标下移一行
86A8    2007AC    JSR    $AC07
86AB    A901      LDA    #$01
86AD    18        CLC
86AE    654D      ADC    $4D
86B0    8509      STA    $09
86B2    A900      LDA    #$00
86B4    654E      ADC    $4E
86B6    850A      STA    $0A
86B8    60        RTS
```

STOP 键扫描处理

```
86B9    A905      LDA    #$05
86BB    8D1640    STA    $4016
86BE    48        PHA
86BF    68        PLA
86C0    48        PHA
86C1    68        PLA
86C2    48        PHA
86C3    68        PLA
86C4    A904      LDA    #$04
86C6    8D1640    STA    $4016
86C9    A906      LDA    #$06
86CB    8D1640    STA    $4016
86CE    AD1740    LDA    $4017
86D1    2910      AND    #$10
86D3    60        RTS
86D4    20B986    JSR    $86B9
86D7    F003      BEQ    $86DC
86D9    60        RTS
```

STOP 保留字入口

```
86DA    68        PLA
86DB    68        PLA
86DC    A900      LDA    #$00
86DE    8563      STA    $63
86E0    2088AF    JSR    $AF88
86E3    D0F7      BNE    $86DC
86E5    20F1B3    JSR    $B3F1
86E8    A50B      LDA    $0B
86EA    8511      STA    $11
86EC    A50C      LDA    $0C
;存当前执行的程序行行号
86EE    8512      STA    $12
86F0    A50D      LDA    $0D
86F2    8513      STA    $13
86F4    A50E      LDA    $0E
;存下一行程序行首址
86F6    8514      STA    $14
86F8    A558      LDA    $58
86FA    8515      STA    $15
86FC    A559      LDA    $59
;存当前执行的指令地址
86FE    8516      STA    $16
8700    A9FF      LDA    #$FF
;置 CONT 有效
8702    8524      STA    $24
8704    C50B      CMP    $0B
;检查当前的程序行行号是否是 HFFFF,若是则 CONT 无效
8706    D007      BNE    $870F
8708    C50C      CMP    $0C
870A    D003      BNE    $870F
870C    203B87    JSR    $873B
870F    4CEAA9    JMP    $A9EA
8712    A90E      LDA    #$0E
;出错处理,类型=CC
8714    4C8FA9    JMP    $A98F
```

CONT 保留字入口

```
8717    A524      LDA    $24
8719    F0F7      BEQ    $8712
871B    A515      LDA    $15
871D    8509      STA    $09
871F    A516      LDA    $16
;取回中断前的指令地址
8721    850A      STA    $0A
;读数
8723    200899    JSR    $9908
8726    207883    JSR    $8378
8729    D0F8      BNE    $8723
;执行 CONT 后,对“:”以后的程序继续处理
872B    A513      LDA    $13
872D    850D      STA    $0D
872F    A514      LDA    $14
8731    850E      STA    $0E
8733    A511      LDA    $11
8735    850B      STA    $0B
8737    A512      LDA    $12
8739    850C      STA    $0C
```

```
873B    A900      LDA    #$00
873D    8524      STA    $24
873F    60        RTS
```

输出操作

PRINT 保留字入口

```
8740    F072      BEQ    $87B4
8742    20829C    JSR    $9C82
8745    A527      LDA    $27
8747    C903      CMP    #$03
;若输出字符串则转$8761
8749    F016      BEQ    $8761
;输出数值变量的处理
874B    AD0004    LDA    $0400
874E    8528      STA    $28
8750    AD0104    LDA    $0401
8753    8529      STA    $29
8755    20968D    JSR    $8D96
;把$3D～$41的内容送入七页
8758    205F86    JSR    $865F
875B    20FB87    JSR    $87FB
875E    4C9487    JMP    $8794
;输出串变量的处理
8761    AD0004    LDA    $0400
8764    F02E      BEQ    $8794
8766    A000      LDY    #$00
8768    AD0104    LDA    $0401
876B    8519      STA    $19
876D    AD0204    LDA    $0402
;取串变量值地址
8770    851A      STA    $1A
8772    B119      LDA    ($19),Y
8774    C920      CMP    #$20
8776    B002      BCS    $877A
8778    A900      LDA    #$00
;取出串变量值存入$500以后的单元
877A    990005    STA    $0500,Y
877D    C8        INY
877E    CE0004    DEC    $0400
8781    D0EF      BNE    $8772
8783    A900      LDA    #$00
8785    990005    STA    $0500,Y
8788    203B8A    JSR    $8A3B
878B    20B787    JSR    $87B7
878E    4C5B87    JMP    $875B
```

PRINT 语句中输出项的处理

```
8791    20A18C    JSR    $8CA1
8794    207883    JSR    $8378
;A=H3A则转$87B4,换行
8797    F01B      BEQ    $87B4
8799    C93B      CMP    #$3B
879B    F009      BEQ    $87A6
879D    C92C      CMP    #$2C
879F    D00D      BNE    $87AE
87A1    A909      LDA    #$09
;是","则调TAB键处理程序
87A3    200DAC    JSR    $AC0D
87A6    207583    JSR    $8375
;是":"则转$87B3,输出结束
87A9    F008      BEQ    $87B3
87AB    4C9487    JMP    $8794
87AE    207883    JSR    $8378
87B1    D08D      BNE    $8740
87B3    60        RTS
;令光标下移一行
87B4    4C07AC    JMP    $AC07
```

输出内容的显示

```
;以$52、$53为地址指针,取数送入七页
87B7    A000      LDY    #$00
87B9    A662      LDX    $62
87BB    B152      LDA    ($52),Y
87BD    F010      BEQ    $87CF
87BF    C920      CMP    #$20
87C1    900F      BCC    $87D2
87C3    9D0007    STA    $0700,X
87C6    C8        INY
87C7    E8        INX
87C8    F003      BEQ    $87CD
87CA    4CBB87    JMP    $87BB
87CD    CA        DEX
87CE    CA        DEX
87CF    8662      STX    $62
87D1    60        RTS
87D2    48        PHA
87D3    8447      STY    $47
87D5    E000      CPX    #$00
87D7    F005      BEQ    $87DE
87D9    8662      STX    $62
87DB    20FB87    JSR    $87FB
```

```
87DE   68       PLA
87DF   200DAC   JSR   $ AC0D
87E2   A900     LDA   # $ 00
87E4   8562     STA   $ 62
87E6   A447     LDY   $ 47
87E8   C8       INY
87E9   A662     LDX   $ 62
87EB   4CBB87   JMP   $ 87BB
87EE   A900     LDA   # $ 00
87F0   A662     LDX   $ 62
87F2   9D0007   STA   $ 0700,X
87F5   E662     INC   $ 62
87F7   60       RTS
;取数送入七页
87F8   20B787   JSR   $ 87B7
87FB   20EE87   JSR   $ 87EE
87FE   A562     LDA   $ 62
8800   F00C     BEQ   $ 880E
8802   A901     LDA   # $ 01
8804   A200     LDX   # $ 00
8806   8664     STX   $ 64
;令 $ 63=1,送显七页的内容
8808   8563     STA   $ 63
880A   A563     LDA   $ 63
880C   D0FC     BNE   $ 880A
880E   60       RTS
    中断处理程序
880F   48       PHA
8810   A900     LDA   # $ 00
8812   8D0320   STA   $ 2003
8815   A902     LDA   # $ 02
8817   8D1440   STA   $ 4014
881A   A5AB     LDA   $ AB
;若启动卡通运动则令 $ AB=HFF
881C   D00E     BNE   $ 882C
881E   A5AC     LDA   $ AC
;执行 MOVE(n)时令 $ AC=HFF
8820   F004     BEQ   $ 8826
8822   A9FF     LDA   # $ FF
8824   85AB     STA   $ AB
8826   A563     LDA   $ 63
8828   D005     BNE   $ 882F
882A   68       PLA
882B   40       RTI
;启动 SP 运动的处理
882C   4C33BE   JMP   $ BE33
    根据 $ 63 的取值,转入相应的处理程序入口
882F   AA       TAX
8830   CA       DEX
8831   8A       TXA
8832   0A       ASL
8833   AA       TAX
8834   BD4188   LDA   $ 8841,X
8837   851B     STA   $ 1B
8839   BD4288   LDA   $ 8842,X
883C   851C     STA   $ 1C
883E   6C1B00   JMP   ( $ 001B)
    系列中断子程序入口地址:
8841        AA 88    25 89    61 89    98 88    70 88    90 89    BE 89    82
            $ 63=1   $ 63=2   $ 63=3   $ 63=4   $ 63=5   $ 63=6   $ 63=7   $ 63=
8850   88   90 88    90 9A    62 88    5F 88    CC 89    02 8A    1A 8A
       8    $ 63=9   $ 63=A   $ 63=B   $ 63=C   $ 63=D   $ 63=E   $ 63=F
;关背景页显示
885F   A900     LDA   # $ 00
;开背景页显示
8861   AEA533   LDX   $ 33A5
8864   8D0120   STA   $ 2001
8867   A900     LDA   # $ 00
8869   8563     STA   $ 63
886B   20E6AB   JSR   $ ABE6
886E   68       PLA
886F   40       RTI
;读出指定 PPU 单元的内容
8870   207688   JSR   $ 8876
;中断返回
8873   4C6788   JMP   $ 8867
;读出 PPU 的内容存入 $ 73 单元
8876   20D8AB   JSR   $ ABD8
8879   AD0720   LDA   $ 2007
887C   AD0720   LDA   $ 2007
887F   8573     STA   $ 73
8881   60       RTS
```

```
8882    207688    JSR    $8876
8885    20D8AB    JSR    $ABD8
8888    A561      LDA    $61
;把光标送入指定的位置显示
888A    8D0720    STA    $2007
888D    4C6788    JMP    $8867
8890    20D8AB    JSR    $ABD8
;把$73的内容送入指定的PPU地址单元
8893    A573      LDA    $73
;中断返回
8895    4C8A88    JMP    $888A
8898    20D8AB    JSR    $ABD8
889B    A648      LDX    $48
889D    A920      LDA    #$20
;清除光标至行尾的所有内容
889F    8D0720    STA    $2007
88A2    E8        INX
88A3    E01C      CPX    #$1C
88A5    D0F8      BNE    $889F
;中断返回
88A7    4C5889    JMP    $8958
;取七页的内容送显
88AA    20A6AB    JSR    $ABA6
88AD    A664      LDX    $64
88AF    20D8AB    JSR    $ABD8
88B2    BD0007    LDA    $0700,X
88B5    F066      BEQ    $891D
88B7    8D0720    STA    $2007
88BA    E648      INC    $48
88BC    A91C      LDA    #$1C
88BE    C548      CMP    $48
88C0    F00F      BEQ    $88D1
88C2    A90E      LDA    #$0E
88C4    C548      CMP    $48
88C6    F003      BEQ    $88CB
88C8    E8        INX
88C9    D0E7      BNE    $88B2
88CB    E8        INX
88CC    8664      STX    $64
;中断返回
88CE    4C5C89    JMP    $895C
;换行处理
88D1    A900      LDA    #$00
88D3    8548      STA    $48
88D5    8A        TXA
88D6    48        PHA
88D7    A649      LDX    $49
88D9    A9FF      LDA    #$FF
88DB    9D0106    STA    $0601,X
88DE    68        PLA
88DF    AA        TAX
88E0    E649      INC    $49
88E2    A918      LDA    #$18
88E4    C549      CMP    $49
88E6    F006      BEQ    $88EE
88E8    E8        INX
88E9    8664      STX    $64
88EB    4C5C89    JMP    $895C
        令屏幕显示上卷一行的处理
88EE    E8        INX
88EF    8664      STX    $64
88F1    208BAD    JSR    $AD8B
88F4    20E6AB    JSR    $ABE6
88F7    A532      LDA    $32
88F9    8D0020    STA    $2000
88FC    20A6AB    JSR    $ABA6
88FF    20FEAA    JSR    $AAFE
8902    20D8AB    JSR    $ABD8
8905    A21C      LDX    #$1C
8907    A920      LDA    #$20
8909    8D0720    STA    $2007
890C    CA        DEX
890D    D0FA      BNE    $8909
890F    8663      STX    $63
8911    20E6AB    JSR    $ABE6
8914    206AAC    JSR    $AC6A
8917    A901      LDA    #$01
8919    8563      STA    $63
891B    68        PLA
891C    40        RTI
;送显字符为零的处理
891D    8563      STA    $63
891F    8562      STA    $62
8921    8564      STA    $64
8923    F037      BEQ    $895C
        把用户本次输入的内容送入五页的处理
8925    A900      LDA    #$00
8927    8548      STA    $48
```

```
8929    20A6AB    JSR    $ABA6
892C    20D8AB    JSR    $ABD8
892F    A01C      LDY    #$1C
8931    AD0720    LDA    $2007
8934    A676      LDX    $76
8936    AD0720    LDA    $2007
8939    9D0005    STA    $0500,X
893C    E8        INX
893D    E0FF      CPX    #$FF
893F    F010      BEQ    $8951
8941    88        DEY
8942    D0F2      BNE    $8936
8944    8676      STX    $76
8946    A649      LDX    $49
8948    BD0106    LDA    $0601,X
894B    F004      BEQ    $8951
894D    E649      INC    $49
894F    D00B      BNE    $895C
8951    A9FF      LDA    #$FF
8953    AEA900    LDX    $00A9
8956    8564      STA    $64
8958    A900      LDA    #$00
895A    8563      STA    $63
895C    20E6AB    JSR    $ABE6
895F    68        PLA
8960    40        RTI
;1. 取出由$4A、$4B指定的
PPU地址的内容送入五页
8961    20D8AB    JSR    $ABD8
8964    AD0720    LDA    $2007
;2. 取出五页的内容存入由
$19、$1A指定的PPU单元中
8967    A200      LDX    #$00
8969    A448      LDY    $48
896B    AD0720    LDA    $2007
896E    9D0005    STA    $0500,X
8971    E8        INX
8972    C8        INY
8973    C01C      CPY    #$1C
8975    D0F4      BNE    $896B
8977    A51A      LDA    $1A
8979    A619      LDX    $19
897B    20F881    JSR    $81F8
897E    A200      LDX    #$00
8980    A448      LDY    $48
8982    BD0005    LDA    $0500,X
8985    8D0720    STA    $2007
8988    E8        INX
8989    C8        INY
898A    C01C      CPY    #$1C
898C    D0F4      BNE    $8982
898E    F0C8      BEQ    $8958
        把背景页的配色数据送入$3FXX
8990    A93F      LDA    #$3F
8992    8D0620    STA    $2006
;$700单元存配色代码$701～$704存颜色代码
8995    AD0007    LDA    $0700
8998    0A        ASL
8999    0A        ASL
899A    8D0620    STA    $2006
899D    AD0107    LDA    $0701
89A0    8D0720    STA    $2007
89A3    AD0207    LDA    $0702
89A6    8D0720    STA    $2007
89A9    AD0307    LDA    $0703
89AC    8D0720    STA    $2007
89AF    AD0407    LDA    $0704
89B2    8D0720    STA    $2007
89B5    20E6AB    JSR    $ABE6
89B8    20F1B3    JSR    $B3F1
89BB    4C6788    JMP    $8867
        把卡通页的配色数据送入$3F××单元
89BE    A93F      LDA    #$3F
89C0    8D0620    STA    $2006
89C3    AD0007    LDA    $0700
89C6    0A        ASL
89C7    0A        ASL
89C8    0910      ORA    #$10
89CA    D0CE      BNE    $899A
89CC    20D8AB    JSR    $ABD8
        执行Ins的操作
89CF    A448      LDY    $48
89D1    A201      LDX    #$01
89D3    A5B0      LDA    $B0
89D5    8D0005    STA    $0500
89D8    AD0720    LDA    $2007
89DB    AD0720    LDA    $2007
89DE    9D0005    STA    $0500,X
```

```
89E1    E8       INX
89E2    C8       INY
89E3    C01C     CPY    # $1C
89E5    D0F4     BNE    $89DB
89E7    85B0     STA    $B0
89E9    20D8AB   JSR    $ABD8
89EC    A200     LDX    # $00
89EE    A448     LDY    $48
89F0    BD0005   LDA    $0500,X
89F3    8D0720   STA    $2007
89F6    E8       INX
89F7    C8       INY
89F8    C01C     CPY    # $1C
89FA    D0F4     BNE    $89F0
89FC    20E6AB   JSR    $ABE6
89FF    4C6788   JMP    $8867
```

;从$4A、$4B定的PPU地址

开始读数,连读H1C个送往五页

```
8A02    20D8AB   JSR    $ABD8
8A05    A200     LDX    # $00
8A07    AD0720   LDA    $2007
8A0A    AD0720   LDA    $2007
8A0D    9D0005   STA    $0500,X
8A10    E8       INX
8A11    E01C     CPX    # $1C
8A13    D0F5     BNE    $8A0A
8A15    A90F     LDA    # $0F
8A17    4C5A89   JMP    $895A
```

;从五页取出一行数据送入PPU

```
8A1A    E649     INC    $49
8A1C    20A6AB   JSR    $ABA6
8A1F    20D8AB   JSR    $ABD8
8A22    A200     LDX    # $00
8A24    BD0005   LDA    $0500,X
8A27    8D0720   STA    $2007
8A2A    E8       INX
8A2B    E01C     CPX    # $1C
8A2D    D0F5     BNE    $8A24
8A2F    4CFC89   JMP    $89FC
```

;令$09、$0A单元指向$500

```
8A32    A900     LDA    # $00
8A34    8509     STA    $09
8A36    A905     LDA    # $05
8A38    850A     STA    $0A
8A3A    60       RTS
```

;令$52、$53单元指向$500

```
8A3B    A900     LDA    # $00
8A3D    8552     STA    $52
8A3F    A905     LDA    # $05
8A41    8553     STA    $53
8A43    60       RTS
```

RAM自由区检查处理程序

```
8A44    A503     LDA    $03
8A46    38       SEC
8A47    E904     SBC    # $04
8A49    852C     STA    $2C
8A4B    A504     LDA    $04
8A4D    E900     SBC    # $00
8A4F    852D     STA    $2D
8A51    20F78B   JSR    $8BF7
8A54    90ED     BCC    $8A43
8A56    A906     LDA    # $06
```

;出错处理,类型=OM

```
8A58    4C8FA9   JMP    $A98F
8A5B    BA       TSX
```

;检查堆栈,若栈顶小于H14则提示内存不足

```
8A5C    E014     CPX    # $14
8A5E    90F6     BCC    $8A56
8A60    60       RTS
```

对$2C、$2D中的数据求补码

```
8A61    38       SEC
8A62    A900     LDA    # $00
8A64    E52C     SBC    $2C
8A66    852C     STA    $2C
8A68    A900     LDA    # $00
8A6A    E52D     SBC    $2D
8A6C    852D     STA    $2D
8A6E    60       RTS
```

SAVE中判断程序量的大小

```
8A6F    A902     LDA    # $02
8A71    8527     STA    $27
8A73    20EFA3   JSR    $A3EF
8A76    207C8A   JSR    $8A7C
8A79    20EC8A   JSR    $8AEC
8A7C    38       SEC
8A7D    A900     LDA    # $00
8A7F    E528     SBC    $28
8A81    8528     STA    $28
```

```
8A83    A900      LDA    #$00
8A85    E529      SBC    $29
8A87    8529      STA    $29
8A89    60        RTS
;出错处理，类型=OV
8A8A    4C348B    JMP    $8B34
8A8D    A52C      LDA    $2C
8A8F    8528      STA    $28
8A91    A52D      LDA    $2D
8A93    8529      STA    $29
8A95    A52E      LDA    $2E
8A97    852A      STA    $2A
8A99    60        RTS
        串变量的处理
8A9A    A621      LDX    $21
8A9C    864A      STX    $4A
8A9E    A528      LDA    $28
8AA0    F0EB      BEQ    $8A8D
8AA2    18        CLC
8AA3    A42C      LDY    $2C
8AA5    F0F2      BEQ    $8A99
8AA7    652C      ADC    $2C
8AA9    C920      CMP    #$20
8AAB    B0DD      BCS    $8A8A
8AAD    854B      STA    $4B
8AAF    A529      LDA    $29
8AB1    8519      STA    $19
8AB3    A52A      LDA    $2A
8AB5    851A      STA    $1A
8AB7    A000      LDY    #$00
8AB9    B119      LDA    ($19),Y
8ABB    9D0007    STA    $0700,X
8ABE    E8        INX
8ABF    C8        INY
8AC0    C628      DEC    $28
8AC2    D0F5      BNE    $8AB9
8AC4    A52D      LDA    $2D
8AC6    8519      STA    $19
8AC8    A52E      LDA    $2E
8ACA    851A      STA    $1A
8ACC    A000      LDY    #$00
8ACE    B119      LDA    ($19),Y
8AD0    9D0007    STA    $0700,X
8AD3    C8        INY
8AD4    E8        INX
8AD5    C62C      DEC    $2C
8AD7    D0F5      BNE    $8ACE
8AD9    8621      STX    $21
8ADB    A54B      LDA    $4B
8ADD    8528      STA    $28
8ADF    A54A      LDA    $4A
8AE1    8529      STA    $29
8AE3    A907      LDA    #$07
8AE5    852A      STA    $2A
8AE7    60        RTS
        数值下标变量的处理
8AE8    A902      LDA    #$02
8AEA    8527      STA    $27
8AEC    A527      LDA    $27
8AEE    C903      CMP    #$03
8AF0    F0A8      BEQ    $8A9A
8AF2    A529      LDA    $29
8AF4    3025      BMI    $8B1B
8AF6    A52D      LDA    $2D
8AF8    3010      BMI    $8B0A
8AFA    18        CLC
8AFB    A52C      LDA    $2C
8AFD    6528      ADC    $28
8AFF    8528      STA    $28
8B01    A52D      LDA    $2D
8B03    6529      ADC    $29
8B05    8529      STA    $29
8B07    302B      BMI    $8B34
8B09    60        RTS
;第二操作数是负数时的处理
8B0A    20618A    JSR    $8A61
8B0D    38        SEC
8B0E    A528      LDA    $28
8B10    E52C      SBC    $2C
8B12    8528      STA    $28
8B14    A529      LDA    $29
8B16    E52D      SBC    $2D
8B18    8529      STA    $29
8B1A    60        RTS
;第一操作数是负数时的处理
8B1B    A52D      LDA    $2D
8B1D    3009      BMI    $8B28
8B1F    207C8A    JSR    $8A7C
```

```
8B22    200D8B    JSR    $8B0D
8B25    4C7C8A    JMP    $8A7C
8B28    207C8A    JSR    $8A7C
8B2B    20618A    JSR    $8A61
8B2E    20FA8A    JSR    $8AFA
8B31    4C7C8A    JMP    $8A7C
8B34    A905      LDA    #$05
;出错处理，类型=OV
8B36    4C8FA9    JMP    $A98F
8B39    18        CLC
8B3A    A52C      LDA    $2C
8B3C    6528      ADC    $28
8B3E    8528      STA    $28
8B40    A52D      LDA    $2D
8B42    6529      ADC    $29
8B44    8529      STA    $29
8B46    60        RTS
8B47    A900      LDA    #$00
8B49    A208      LDX    #$08
;令$35～$3C=0
8B4B    9534      STA    $34,X
8B4D    CA        DEX
8B4E    D0FB      BNE    $8B4B
8B50    60        RTS
        乘法运算子程序
8B51    20598B    JSR    $8B59
8B54    A536      LDA    $36
8B56    D0DC      BNE    $8B34
8B58    60        RTS
8B59    20EFA3    JSR    $A3EF
8B5C    20478B    JSR    $8B47
8B5F    20928B    JSR    $8B92
8B62    48        PHA
8B63    A210      LDX    #$10
8B65    18        CLC
8B66    6629      ROR    $29
8B68    6628      ROR    $28
8B6A    900D      BCC    $8B79
8B6C    18        CLC
;乘法运算处理被乘数存$28、$29单元 乘数
存$2C、$2D单元 乘积存$28、$29单元
8B6D    A536      LDA    $36
8B6F    652C      ADC    $2C
8B71    8536      STA    $36
8B73    A537      LDA    $37
8B75    652D      ADC    $2D
8B77    8537      STA    $37
8B79    6637      ROR    $37
8B7B    6636      ROR    $36
8B7D    6635      ROR    $35
8B7F    6634      ROR    $34
8B81    CA        DEX
8B82    D0E2      BNE    $8B66
8B84    A534      LDA    $34
8B86    8528      STA    $28
8B88    A535      LDA    $35
8B8A    8529      STA    $29
8B8C    68        PLA
8B8D    F022      BEQ    $8BB1
8B8F    4C7C8A    JMP    $8A7C
8B92    A900      LDA    #$00
8B94    8534      STA    $34
8B96    A529      LDA    $29
8B98    1007      BPL    $8BA1
8B9A    207C8A    JSR    $8A7C
8B9D    A9FF      LDA    #$FF
8B9F    8534      STA    $34
8BA1    A52D      LDA    $2D
8BA3    100A      BPL    $8BAF
8BA5    20618A    JSR    $8A61
8BA8    A9FF      LDA    #$FF
8BAA    38        SEC
8BAB    E534      SBC    $34
8BAD    8534      STA    $34
8BAF    A534      LDA    $34
8BB1    60        RTS
        除法运算子程序
8BB2    20EFA3    JSR    $A3EF
8BB5    20928B    JSR    $8B92
8BB8    48        PHA
8BB9    A900      LDA    #$00
8BBB    852A      STA    $2A
8BBD    852B      STA    $2B
8BBF    A210      LDX    #$10
8BC1    0628      ASL    $28
8BC3    2629      ROL    $29
8BC5    262A      ROL    $2A
8BC7    262B      ROL    $2B
```

;除法运算处理被除数存 $ 28、$ 29 单元 除数存 $ 2C、$ 2D 单元 商存 $ 28、$ 29 单元

```
8BC9    A52A      LDA   $2A
8BCB    38        SEC
8BCC    E52C      SBC   $2C
8BCE    A52B      LDA   $2B
8BD0    E52D      SBC   $2D
8BD2    900F      BCC   $8BE3
8BD4    A52A      LDA   $2A
8BD6    38        SEC
8BD7    E52C      SBC   $2C
8BD9    852A      STA   $2A
8BDB    A52B      LDA   $2B
8BDD    E52D      SBC   $2D
8BDF    852B      STA   $2B
8BE1    E628      INC   $28
8BE3    CA        DEX
8BE4    D0DB      BNE   $8BC1
8BE6    4C8C8B    JMP   $8B8C
8BE9    38        SEC
8BEA    A528      LDA   $28
8BEC    E52C      SBC   $2C
8BEE    8528      STA   $28
8BF0    A529      LDA   $29
8BF2    E52D      SBC   $2D
8BF4    8529      STA   $29
8BF6    60        RTS
8BF7    A529      LDA   $29
8BF9    C52D      CMP   $2D
8BFB    F001      BEQ   $8BFE
8BFD    60        RTS
8BFE    A528      LDA   $28
8C00    C52C      CMP   $2C
8C02    60        RTS
```

比较 $ 28、$ 29 与 $ 2C、$ 2D 中两数的大小,设定不同的标志

```
8C03    A527      LDA   $27
8C05    C903      CMP   #$03
8C07    F041      BEQ   $8C4A
```

;若 $ 28、$ 29 = $ 2C、$ 2D 则令 C=1、A=0

```
8C09    A529      LDA   $29
8C0B    3023      BMI   $8C30
```

;若 $ 28、$ 29 > $ 2C、$ 2D 则令 C=1、A=HFF

若 $ 28、$ 29 < $ 2C、$ 2D 则令 C=0、A=HFF

```
8C0D    A52D      LDA   $2D
8C0F    300C      BMI   $8C1D
8C11    A529      LDA   $29
8C13    C52D      CMP   $2D
8C15    F00A      BEQ   $8C21
8C17    B004      BCS   $8C1D
8C19    18        CLC
8C1A    A9FF      LDA   #$FF
8C1C    60        RTS
8C1D    38        SEC
8C1E    A9FF      LDA   #$FF
8C20    60        RTS
8C21    A528      LDA   $28
8C23    C52C      CMP   $2C
8C25    F005      BEQ   $8C2C
8C27    B0F4      BCS   $8C1D
8C29    4C198C    JMP   $8C19
8C2C    38        SEC
8C2D    A900      LDA   #$00
8C2F    60        RTS
8C30    A52D      LDA   $2D
8C32    10E5      BPL   $8C19
8C34    A529      LDA   $29
8C36    C52D      CMP   $2D
8C38    F005      BEQ   $8C3F
8C3A    B0E1      BCS   $8C1D
8C3C    4C198C    JMP   $8C19
8C3F    A528      LDA   $28
8C41    C52C      CMP   $2C
8C43    F0E7      BEQ   $8C2C
8C45    B0D6      BCS   $8C1D
8C47    4C198C    JMP   $8C19
```

IF A $ =... 串变量的处理

```
8C4A    A529      LDA   $29
8C4C    8519      STA   $19
8C4E    A52A      LDA   $2A
8C50    851A      STA   $1A
8C52    A52D      LDA   $2D
8C54    851B      STA   $1B
8C56    A52E      LDA   $2E
```

```
8C58    851C      STA   $1C
8C5A    A528      LDA   $28
8C5C    F028      BEQ   $8C86
8C5E    854A      STA   $4A
8C60    F0BB      BEQ   $8C1D
8C62    A52C      LDA   $2C
8C64    854B      STA   $4B
8C66    A000      LDY   #$00
8C68    B119      LDA   ($19),Y
8C6A    D11B      CMP   ($1B),Y
8C6C    90AB      BCC   $8C19
8C6E    F003      BEQ   $8C73
8C70    4C1D8C    JMP   $8C1D
8C73    C8        INY
8C74    C64A      DEC   $4A
8C76    F007      BEQ   $8C7F
8C78    C64B      DEC   $4B
8C7A    F0A1      BEQ   $8C1D
8C7C    4C688C    JMP   $8C68
8C7F    C64B      DEC   $4B
8C81    F0A9      BEQ   $8C2C
8C83    4C198C    JMP   $8C19
8C86    A52C      LDA   $2C
8C88    F0A2      BEQ   $8C2C
8C8A    4C198C    JMP   $8C19
8C8D    A000      LDY   #$00
8C8F    9109      STA   ($09),Y
8C91    E609      INC   $09
8C93    F001      BEQ   $8C96
8C95    60        RTS
8C96    C609      DEC   $09
8C98    A900      LDA   #$00
8C9A    9109      STA   ($09),Y
8C9C    60        RTS
8C9D    A000      LDY   #$00
8C9F    B109      LDA   ($09),Y
8CA1    E609      INC   $09
8CA3    F003      BEQ   $8CA8
8CA5    29FF      AND   #$FF
8CA7    60        RTS
8CA8    E60A      INC   $0A
8CAA    29FF      AND   #$FF
8CAC    60        RTS
;空格的翻译
8CAD    A000      LDY   #$00
8CAF    E609      INC   $09
8CB1    F003      BEQ   $8CB6
8CB3    B109      LDA   ($09),Y
8CB5    60        RTS
8CB6    E60A      INC   $0A
8CB8    B109      LDA   ($09),Y
8CBA    60        RTS
8CBB    A509      LDA   $09
8CBD    F003      BEQ   $8CC2
8CBF    C609      DEC   $09
8CC1    60        RTS
8CC2    C609      DEC   $09
8CC4    C60A      DEC   $0A
8CC6    60        RTS
8CC7    A000      LDY   #$00
8CC9    B109      LDA   ($09),Y
8CCB    60        RTS
8CCC    A000      LDY   #$00
8CCE    9109      STA   ($09),Y
8CD0    60        RTS
8CD1    20A18C    JSR   $8CA1
8CD4    A000      LDY   #$00
8CD6    B109      LDA   ($09),Y
8CD8    C920      CMP   #$20
8CDA    F0F5      BEQ   $8CD1
8CDC    60        RTS
8CDD    C930      CMP   #$30
8CDF    9006      BCC   $8CE7
8CE1    C93A      CMP   #$3A
8CE3    B002      BCS   $8CE7
8CE5    18        CLC
8CE6    60        RTS
8CE7    38        SEC
8CE8    60        RTS
        对输出数据整形,删除数据前的“0”
8CE9    E646      INC   $46
8CEB    A646      LDX   $46
8CED    B53D      LDA   $3D,X
8CEF    F005      BEQ   $8CF6
8CF1    C930      CMP   #$30
8CF3    F0F4      BEQ   $8CE9
8CF5    60        RTS
8CF6    C646      DEC   $46
```

```
8CF8    60        RTS
8CF9    20D48C    JSR     $8CD4
8CFC    C92D      CMP     #$2D
8CFE    D009      BNE     $8D09
8D00    20A18C    JSR     $8CA1
8D03    20F98C    JSR     $8CF9
8D06    4C7C8A    JMP     $8A7C
```

把以 ASCII 码表示的十进制数转换为十六进制数存入 $28、$29

```
8D09    A902      LDA     #$02
8D0B    8527      STA     $27
8D0D    A900      LDA     #$00
8D0F    8528      STA     $28
8D11    8529      STA     $29
8D13    A90A      LDA     #$0A
8D15    852C      STA     $2C
8D17    A900      LDA     #$00
8D19    852D      STA     $2D
8D1B    A000      LDY     #$00
8D1D    B109      LDA     ($09),Y
8D1F    20DD8C    JSR     $8CDD
8D22    B0C4      BCS     $8CE8
8D24    48        PHA
8D25    20518B    JSR     $8B51
8D28    68        PLA
8D29    38        SEC
8D2A    E930      SBC     #$30
8D2C    852C      STA     $2C
8D2E    20398B    JSR     $8B39
8D31    B006      BCS     $8D39
8D33    20A18C    JSR     $8CA1
8D36    4C138D    JMP     $8D13
8D39    4C348B    JMP     $8B34
```

把十六进制数据转换为十进制的 ASCII 码存入 $3D～$41

```
8D3C    A900      LDA     #$00
8D3E    8546      STA     $46
8D40    A546      LDA     $46
8D42    48        PHA
8D43    A910      LDA     #$10
8D45    852C      STA     $2C
8D47    A927      LDA     #$27
8D49    852D      STA     $2D
;转换第一位，万位
8D4B    20888D    JSR     $8D88
8D4E    A9E8      LDA     #$E8
8D50    852C      STA     $2C
8D52    A903      LDA     #$03
8D54    852D      STA     $2D
;转换第二位，千位
8D56    20888D    JSR     $8D88
8D59    A964      LDA     #$64
8D5B    852C      STA     $2C
8D5D    A900      LDA     #$00
8D5F    852D      STA     $2D
;转换第三位，百位
8D61    20888D    JSR     $8D88
8D64    A90A      LDA     #$0A
8D66    852C      STA     $2C
;转换第四位，十位
8D68    20888D    JSR     $8D88
8D6B    A528      LDA     $28
8D6D    18        CLC
8D6E    6930      ADC     #$30
;转换第五位，个位
8D70    20818D    JSR     $8D81
8D73    A900      LDA     #$00
8D75    20818D    JSR     $8D81
8D78    68        PLA
8D79    8546      STA     $46
8D7B    60        RTS
8D7C    20398B    JSR     $8B39
8D7F    A547      LDA     $47
8D81    A646      LDX     $46
8D83    953D      STA     $3D,X
8D85    E646      INC     $46
8D87    60        RTS
8D88    A930      LDA     #$30
8D8A    8547      STA     $47
8D8C    20E98B    JSR     $8BE9
8D8F    90EB      BCC     $8D7C
8D91    E647      INC     $47
8D93    4C8C8D    JMP     $8D8C
```

PRINT 输出数据的检查处理

```
8D96    A900      LDA     #$00
8D98    8546      STA     $46
8D9A    A020      LDY     #$20
8D9C    A529      LDA     $29
```

;若是正数则直接转换为 ASCII 码

若是负数则先置负号,再把它转换为 ASCII 码

```
8D9E    1005      BPL    $8DA5
8DA0    207C8A    JSR    $8A7C
8DA3    A02D      LDY    #$2D
8DA5    98        TYA
8DA6    48        PHA
8DA7    20408D    JSR    $8D40
8DAA    20EB8C    JSR    $8CEB
8DAD    C646      DEC    $46
8DAF    68        PLA
8DB0    20818D    JSR    $8D81
8DB3    C646      DEC    $46
8DB5    60        RTS
```

;把数据转换为 ASCII 码子程序

```
8DB6    C90A      CMP    #$0A
8DB8    B003      BCS    $8DBD
8DBA    6930      ADC    #$30
8DBC    60        RTS
8DBD    18        CLC
8DBE    6937      ADC    #$37
8DC0    60        RTS
```

把一个单元的十六进制数据按高、低位

转换为 ASCII 码存入 $3D～$40 单元

```
8DC1    48        PHA
8DC2    29F0      AND    #$F0
8DC4    4A        LSR
8DC5    4A        LSR
8DC6    4A        LSR
8DC7    4A        LSR
8DC8    20B68D    JSR    $8DB6
8DCB    953D      STA    $3D,X
8DCD    E8        INX
8DCE    68        PLA
8DCF    290F      AND    #$0F
8DD1    20B68D    JSR    $8DB6
8DD4    953D      STA    $3D,X
8DD6    E8        INX
8DD7    60        RTS
```

把 $28、$29 中的十六进制数转换为

四个 ASCII 码存入 $3D、$41 单元

```
8DD8    A200      LDX    #$00
8DDA    8646      STX    $46
8DDC    A529      LDA    $29
8DDE    20C18D    JSR    $8DC1
8DE1    A528      LDA    $28
8DE3    20C18D    JSR    $8DC1
8DE6    A900      LDA    #$00
8DE8    953D      STA    $3D,X
8DEA    60        RTS
```

十六进制数据的翻译

```
8DEB    A902      LDA    #$02
8DED    8527      STA    $27
8DEF    A200      LDX    #$00
8DF1    A900      LDA    #$00
8DF3    8528      STA    $28
8DF5    8529      STA    $29
```

;读数,检查 16 进制数据是否合法

若 A<H30 则令 C=0

若 A<H3A 则转 $8E0E,翻译结束

若 A<H41 则令 C=0

若 A>H46 则令 C=1 若合法则入栈

```
8DF7    20C78C    JSR    $8CC7
8DFA    38        SEC
8DFB    E930      SBC    #$30
8DFD    90EB      BCC    $8DEA
8DFF    C90A      CMP    #$0A
8E01    900B      BCC    $8E0E
8E03    38        SEC
8E04    E907      SBC    #$07
8E06    C90A      CMP    #$0A
8E08    90E0      BCC    $8DEA
8E0A    C910      CMP    #$10
8E0C    B0DC      BCS    $8DEA
8E0E    48        PHA
8E0F    20228E    JSR    $8E22
8E12    68        PLA
8E13    852C      STA    $2C
8E15    A900      LDA    #$00
8E17    852D      STA    $2D
8E19    20398B    JSR    $8B39
8E1C    20A18C    JSR    $8CA1
8E1F    4CF78D    JMP    $8DF7
8E22    8A        TXA
8E23    48        PHA
8E24    A910      LDA    #$10
8E26    852C      STA    $2C
8E28    A900      LDA    #$00
```

```
8E2A    852D      STA   $2D
8E2C    20518B    JSR   $8B51
8E2F    68        PLA
8E30    AA        TAX
8E31    60        RTS
8E32    203E8E    JSR   $8E3E
8E35    A000      LDY   #$00
8E37    B14D      LDA   ($4D),Y
8E39    60        RTS
8E3A    A000      LDY   #$00
8E3C    B14D      LDA   ($4D),Y
8E3E    E64D      INC   $4D
8E40    D011      BNE   $8E53
8E42    E64E      INC   $4E
8E44    D00D      BNE   $8E53
    翻译结束处理
8E46    A900      LDA   #$00
8E48    204B8E    JSR   $8E4B
8E4B    A000      LDY   #$00
8E4D    914D      STA   ($4D),Y
8E4F    E64D      INC   $4D
;若$4D=0则表示一行字符超过HFF个，
转出错处理,类型=OV,
8E51    F003      BEQ   $8E56
8E53    29FF      AND   #$FF
8E55    60        RTS
8E56    4C348B    JMP   $8B34
;翻译程序后送入三页
8E59    A900      LDA   #$00
8E5B    854D      STA   $4D
8E5D    A903      LDA   #$03
8E5F    854E      STA   $4E
8E61    20FA8E    JSR   $8EFA
8E64    29FF      AND   #$FF
8E66    F0DE      BEQ   $8E46
8E68    C93F      CMP   #$3F
;是问号转$8EAD
8E6A    F041      BEQ   $8EAD
8E6C    C922      CMP   #$22
;是引号转$8EB6
8E6E    F046      BEQ   $8EB6
8E70    C927      CMP   #$27
;是单引号转$8EB3
8E72    F03F      BEQ   $8EB3
8E74    C92E      CMP   #$2E
;是点号转$8E86
8E76    F00E      BEQ   $8E86
8E78    C926      CMP   #$26
;是"&"转$8ED5
8E7A    F059      BEQ   $8ED5
8E7C    20DD8C    JSR   $8CDD
8E7F    904B      BCC   $8ECC
;检索保留字
8E81    20AE8F    JSR   $8FAE
8E84    902A      BCC   $8EB0
8E86    20A18C    JSR   $8CA1
8E89    C980      CMP   #$80
;是保留字转$8E61
8E8B    B0D4      BCS   $8E61
8E8D    C920      CMP   #$20
8E8F    90D0      BCC   $8E61
8E91    C940      CMP   #$40
8E93    9012      BCC   $8EA7
8E95    204B8E    JSR   $8E4B
8E98    A000      LDY   #$00
8E9A    B109      LDA   ($09),Y
8E9C    20BBA1    JSR   $A1BB
8E9F    B0C0      BCS   $8E61
8EA1    20A18C    JSR   $8CA1
8EA4    4C958E    JMP   $8E95
8EA7    204B8E    JSR   $8E4B
8EAA    4C618E    JMP   $8E61
"?"的翻译
8EAD    4C248F    JMP   $8F24
8EB0    4C2B8F    JMP   $8F2B
    读到引号的翻译
8EB3    A000      LDY   #$00
8EB5    AEA022    LDX   $22A0
8EB8    8499      STY   $99
8EBA    204B8E    JSR   $8E4B
8EBD    20A18C    JSR   $8CA1
8EC0    20C78C    JSR   $8CC7
8EC3    F0A1      BEQ   $8E66
8EC5    C599      CMP   $99
8EC7    D0F1      BNE   $8EBA
8EC9    4C868E    JMP   $8E86
```

把由键盘输入的十进制数据转换为十六进制数据存入$28、$29

```
8ECC    A912      LDA    # $12
8ECE    48        PHA
8ECF    20098D    JSR    $8D09
8ED2    4CE88E    JMP    $8EE8
```

读到"&"的翻译

```
8ED5    20A18C    JSR    $8CA1
8ED8    20C78C    JSR    $8CC7
8EDB    C948      CMP    # $48
8EDD    D003      BNE    $8EE2
8EDF    20A18C    JSR    $8CA1
8EE2    A911      LDA    # $11
8EE4    48        PHA
8EE5    20EB8D    JSR    $8DEB
8EE8    68        PLA
8EE9    204B8E    JSR    $8E4B
8EEC    A528      LDA    $28
8EEE    204B8E    JSR    $8E4B
8EF1    A529      LDA    $29
8EF3    204B8E    JSR    $8E4B
8EF6    4C618E    JMP    $8E61
8EF9    60        RTS
8EFA    20C78C    JSR    $8CC7
8EFD    C920      CMP    # $20
8EFF    D0F8      BNE    $8EF9
8F01    204B8E    JSR    $8E4B
8F04    20A18C    JSR    $8CA1
8F07    4CFA8E    JMP    $8EFA
;是 REM 的处理
8F0A    A000      LDY    # $00
8F0C    8599      STA    $99
8F0E    4CC08E    JMP    $8EC0
;是 LIST 的处理
8F11    20FA8E    JSR    $8EFA
8F14    20DD8C    JSR    $8CDD
8F17    902F      BCC    $8F48
8F19    C92C      CMP    # $2C
8F1B    F048      BEQ    $8F65
8F1D    C92D      CMP    # $2D
8F1F    F044      BEQ    $8F65
8F21    4C618E    JMP    $8E61
```

读到"?"的直接翻译

```
8F24    A98B      LDA    # $8B
8F26    8599      STA    $99
8F28    20A18C    JSR    $8CA1
8F2B    A599      LDA    $99
8F2D    204B8E    JSR    $8E4B
8F30    C995      CMP    # $95
;是 REM 转 $8F0A
8F32    F0D6      BEQ    $8F0A
8F34    C986      CMP    # $86
;是 LIST 转 $8F11
8F36    F0D9      BEQ    $8F11
8F38    C991      CMP    # $91
;是 DATA 转 $8F76
8F3A    F03A      BEQ    $8F76
8F3C    C987      CMP    # $87
8F3E    B0E1      BCS    $8F21
8F40    20FA8E    JSR    $8EFA
8F43    20DD8C    JSR    $8CDD
8F46    B0D9      BCS    $8F21
8F48    20098D    JSR    $8D09
8F4B    A90B      LDA    # $0B
8F4D    204B8E    JSR    $8E4B
8F50    A528      LDA    $28
8F52    204B8E    JSR    $8E4B
8F55    A529      LDA    $29
8F57    204B8E    JSR    $8E4B
8F5A    20FA8E    JSR    $8EFA
8F5D    C92C      CMP    # $2C
8F5F    F004      BEQ    $8F65
8F61    C92D      CMP    # $2D
8F63    D0BC      BNE    $8F21
```

标点符号的翻译

```
8F65    204B8E    JSR    $8E4B
8F68    20A18C    JSR    $8CA1
8F6B    20FA8E    JSR    $8EFA
8F6E    20DD8C    JSR    $8CDD
8F71    90D5      BCC    $8F48
8F73    4C5D8F    JMP    $8F5D
```

DATA 语句的翻译

```
8F76    20FA8E    JSR    $8EFA
8F79    C93A      CMP    # $3A
8F7B    F0A4      BEQ    $8F21
8F7D    29FF      AND    # $FF
8F7F    F0A0      BEQ    $8F21
8F81    C922      CMP    # $22
8F83    F009      BEQ    $8F8E
8F85    204B8E    JSR    $8E4B
```

```
8F88    20A18C    JSR    $8CA1
8F8B    4C768F    JMP    $8F76
8F8E    204B8E    JSR    $8E4B
8F91    20A18C    JSR    $8CA1
8F94    20FA8E    JSR    $8EFA
8F97    C922      CMP    #$22
8F99    F0EA      BEQ    $8F85
8F9B    29FF      AND    #$FF
8F9D    F0E0      BEQ    $8F7F
8F9F    4C8E8F    JMP    $8F8E
8FA2    A000      LDY    #$00
8FA4    B154      LDA    ($54),Y
8FA6    60        RTS
8FA7    E654      INC    $54
8FA9    D0FB      BNE    $8FA6
8FAB    E655      INC    $55
8FAD    60        RTS
```

保留字的检索与翻译

```
8FAE    A928      LDA    #$28
8FB0    8554      STA    $54
8FB2    A9C1      LDA    #$C1
8FB4    8555      STA    $55
8FB6    20A28F    JSR    $8FA2
8FB9    AA        TAX
8FBA    E8        INX
8FBB    F036      BEQ    $8FF3
8FBD    8599      STA    $99
8FBF    20A78F    JSR    $8FA7
8FC2    20F88F    JSR    $8FF8
```

;当检索到第一个字母相同时，再比较以后的字母，直到不同时转出

```
8FC5    20C78C    JSR    $8CC7
8FC8    8598      STA    $98
8FCA    20A28F    JSR    $8FA2
8FCD    C598      CMP    $98
8FCF    D00D      BNE    $8FDE
8FD1    20A18C    JSR    $8CA1
8FD4    20A78F    JSR    $8FA7
8FD7    20A28F    JSR    $8FA2
8FDA    10E9      BPL    $8FC5
8FDC    18        CLC
8FDD    60        RTS
8FDE    209D8C    JSR    $8C9D
8FE1    C92E      CMP    #$2E
8FE3    F0F7      BEQ    $8FDC
8FE5    200190    JSR    $9001
```

;从保留字表中逐个查找与五页的指令相同的保留字代码

```
8FE8    20A28F    JSR    $8FA2
8FEB    30C9      BMI    $8FB6
8FED    20A78F    JSR    $8FA7
8FF0    4CE88F    JMP    $8FE8
8FF3    20C78C    JSR    $8CC7
8FF6    38        SEC
8FF7    60        RTS
8FF8    A509      LDA    $09
8FFA    859A      STA    $9A
8FFC    A50A      LDA    $0A
8FFE    859B      STA    $9B
9000    60        RTS
9001    A59A      LDA    $9A
9003    8509      STA    $09
9005    A59B      LDA    $9B
9007    850A      STA    $0A
9009    60        RTS
```

LIST 语句执行中，程序体的翻译、送显处理

```
900A    20328A    JSR    $8A32
900D    20358E    JSR    $8E35
9010    F017      BEQ    $9029
9012    C920      CMP    #$20
9014    9018      BCC    $902E
9016    C980      CMP    #$80
9018    904A      BCC    $9064
901A    48        PHA
901B    209490    JSR    $9094
901E    203E8E    JSR    $8E3E
9021    68        PLA
9022    C995      CMP    #$95
9024    F060      BEQ    $9086
9026    4C0D90    JMP    $900D
9029    A900      LDA    #$00
902B    4C8D8C    JMP    $8C8D
```

;LIST 中读数小于 H20 的处理

```
902E    48        PHA
902F    20328E    JSR    $8E32
9032    8528      STA    $28
9034    20328E    JSR    $8E32
9037    8529      STA    $29
```

```
9039    203E8E    JSR    $8E3E
903C    68        PLA
903D    C911      CMP    #$11
903F    D010      BNE    $9051
9041    A926      LDA    #$26
;读到十六进制时送显"&"的处理
9043    208D8C    JSR    $8C8D
9046    A948      LDA    #$48
;送显"H"的处理
9048    208D8C    JSR    $8C8D
904B    20D88D    JSR    $8DD8
904E    4C5490    JMP    $9054
;读到十进制数的处理
9051    203C8D    JSR    $8D3C
9054    20EB8C    JSR    $8CEB
9057    A646      LDX    $46
9059    B53D      LDA    $3D,X
905B    F0B0      BEQ    $900D
905D    208D8C    JSR    $8C8D
9060    E8        INX
9061    4C5990    JMP    $9059
;从程序区读数小于H80的处理
9064    C922      CMP    #$22
9066    D017      BNE    $907F
;双引号中内容的读出处理
9068    208E90    JSR    $908E
906B    20358E    JSR    $8E35
906E    C922      CMP    #$22
9070    F007      BEQ    $9079
9072    29FF      AND    #$FF
9074    D0F2      BNE    $9068
9076    4C0D90    JMP    $900D
9079    208E90    JSR    $908E
907C    4C0D90    JMP    $900D
907F    C927      CMP    #$27
9081    D0F6      BNE    $9079
;读到单引号时的处理
9083    208E90    JSR    $908E
9086    20358E    JSR    $8E35
9089    D0F8      BNE    $9083
908B    4C0D90    JMP    $900D
908E    203A8E    JSR    $8E3A
9091    4C8D8C    JMP    $8C8D
;LIST中保留字的翻译处理
9094    8599      STA    $99
9096    A928      LDA    #$28
9098    8554      STA    $54
909A    A9C1      LDA    #$C1
909C    8555      STA    $55
;读出保留字代码
909E    20A28F    JSR    $8FA2
90A1    C9FF      CMP    #$FF
90A3    F020      BEQ    $90C5
90A5    C599      CMP    $99
90A7    F0B       BEQ    $90B4
90A9    20A78F    JSR    $8FA7
90AC    20A28F    JSR    $8FA2
90AF    10F8      BPL    $90A9
90B1    4CA190    JMP    $90A1
;把保留字的内容送入五页
90B4    20A78F    JSR    $8FA7
90B7    20A28F    JSR    $8FA2
90BA    3009      BMI    $90C5
90BC    208D8C    JSR    $8C8D
90BF    20A78F    JSR    $8FA7
90C2    4CB790    JMP    $90B7
90C5    60        RTS
```

GOSUB 保留字入口

```
;无行号则出错
90C6    F05B      BEQ    $9123
90C8    AA        TAX
90C9    68        PLA
90CA    855F      STA    $5F
90CC    68        PLA
90CD    8560      STA    $60
90CF    8A        TXA
;检索目标行号
90D0    207291    JSR    $9172
90D3    209895    JSR    $9598
;保护转子前的现场
90D6    A509      LDA    $09
90D8    48        PHA
90D9    A50A      LDA    $0A
90DB    48        PHA
90DC    A50B      LDA    $0B
90DE    48        PHA
90DF    A50C      LDA    $0C
90E1    48        PHA
```

```
90E2    A50D      LDA    $0D
90E4    48        PHA
90E5    A50E      LDA    $0E
90E7    48        PHA
90E8    A900      LDA    #$00
90EA    48        PHA
90EB    A9FF      LDA    #$FF
90ED    48        PHA
90EE    A560      LDA    $60
90F0    48        PHA
90F1    A55F      LDA    $5F
90F3    48        PHA
90F4    4C4B91    JMP    $914B
```

RETURN 后无行号的处理

```
90F7    A902      LDA    #$02
;出错处理,类型=RG
90F9    4C8FA9    JMP    $A98F
;转子前的现场恢复
90FC    68        PLA
90FD    855F      STA    $5F
90FF    68        PLA
9100    8560      STA    $60
9102    68        PLA
9103    C9FF      CMP    #$FF
9105    D0F0      BNE    $90F7
9107    68        PLA
9108    D0ED      BNE    $90F7
910A    68        PLA
910B    850E      STA    $0E
910D    68        PLA
910E    850D      STA    $0D
9110    68        PLA
9111    850C      STA    $0C
9113    68        PLA
9114    850B      STA    $0B
9116    68        PLA
9117    850A      STA    $0A
9119    68        PLA
911A    8509      STA    $09
911C    A560      LDA    $60
911E    48        PHA
911F    A55F      LDA    $5F
9121    48        PHA
9122    60        RTS
;出错处理,类型=SN
9123    4C6D83    JMP    $836D
```

RETURN 保留字入口

```
;RETURN 后无行号时转$90FC
9126    F0D4      BEQ    $90FC
9128    68        PLA
9129    855F      STA    $5F
912B    68        PLA
912C    8560      STA    $60
912E    68        PLA
912F    C9FF      CMP    #$FF
9131    D0C4      BNE    $90F7
9133    68        PLA
9134    D0C1      BNE    $90F7
9136    BA        TSX
9137    8A        TXA
9138    18        CLC
9139    6906      ADC    #$06
913B    AA        TAX
913C    9A        TXS
913D    A560      LDA    $60
913F    48        PHA
9140    A55F      LDA    $5F
9142    48        PHA
9143    207883    JSR    $8378
```

GOTO 保留字入口

```
;GOTO 后无行号则出错
9146    F0DB      BEQ    $9123
;检索目标行号
9148    207291    JSR    $9172
914B    A000      LDY    #$00
914D    B15B      LDA    ($5B),Y
914F    18        CLC
9150    655B      ADC    $5B
9152    850D      STA    $0D
9154    A900      LDA    #$00
9156    655C      ADC    $5C
9158    850E      STA    $0E
915A    C8        INY
915B    B15B      LDA    ($5B),Y
915D    850B      STA    $0B
915F    C8        INY
9160    B15B      LDA    ($5B),Y
9162    850C      STA    $0C
```

```
9164    18       CLC
9165    A903     LDA    #$03
9167    655B     ADC    $5B
9169    8509     STA    $09
916B    A900     LDA    #$00
916D    655C     ADC    $5C
916F    850A     STA    $0A
9171    60       RTS
```

GOTO 语句中检索转向行号的处理

```
9172    C90B     CMP    #$0B
9174    D0AD     BNE    $9123
9176    A001     LDY    #$01
9178    B109     LDA    ($09),Y
917A    8544     STA    $44
917C    C8       INY
917D    B109     LDA    ($09),Y
917F    8545     STA    $45
9181    200484   JSR    $8404
9184    D001     BNE    $9187
9186    60       RTS
9187    A907     LDA    #$07
;出错处理,类型=UL
9189    4C8FA9   JMP    $A98F
```

CALL 保留字入口

```
918C    20829C   JSR    $9C82
918F    6C0004   JMP    ($0400)
;出错处理,类型=IL
9192    4C7083   JMP    $8370
9195    4C63B2   JMP    $B263
```

PAUSE 保留字入口

```
9198    F0FB     BEQ    $9195
919A    A900     LDA    #$00
919C    8574     STA    $74
919E    20829C   JSR    $9C82
91A1    20EFA3   JSR    $A3EF
91A4    AD0104   LDA    $0401
91A7    30E9     BMI    $9192
91A9    D005     BNE    $91B0
91AB    AD0004   LDA    $0400
91AE    F0D6     BEQ    $9186
91B0    EE0104   INC    $0401
91B3    20D486   JSR    $86D4
;以$74单元为第一延时器
91B6    C674     DEC    $74
;以$400单元为第二延时器
91B8    D0F9     BNE    $91B3
;以$401单元为第三延时器
91BA    CE0004   DEC    $0400
91BD    D0F4     BNE    $91B3
91BF    AD0104   LDA    $0401
91C2    F005     BEQ    $91C9
91C4    CE0104   DEC    $0401
91C7    D0EA     BNE    $91B3
91C9    60       RTS
```

LOCATE 保留字入口

```
91CA    206D9C   JSR    $9C6D
91CD    C91C     CMP    #$1C
91CF    B0C1     BCS    $9192
;X 座标值存入$6B单元
91D1    856B     STA    $6B
91D3    206A9C   JSR    $9C6A
91D6    C918     CMP    #$18
91D8    B0B8     BCS    $9192
;Y 座标值存入$6C单元
91DA    856C     STA    $6C
91DC    4CAAAC   JMP    $ACAA
```

POKE 保留字入口

```
;写入地址存入$400、$401单元
91DF    20829C   JSR    $9C82
91E2    AD0004   LDA    $0400
91E5    8525     STA    $25
91E7    AD0104   LDA    $0401
91EA    8526     STA    $26
91EC    A000     LDY    #$00
91EE    8446     STY    $46
91F0    206A9C   JSR    $9C6A
91F3    A446     LDY    $46
;把数据写入指定地址
91F5    9125     STA    ($25),Y
91F7    E646     INC    $46
91F9    207883   JSR    $8378
91FC    D0F2     BNE    $91F0
91FE    60       RTS
91FF    68       PLA
9200    68       PLA
9201    4C4583   JMP    $8345
```

IF 保留字入口

```
9204    20829C   JSR    $9C82
```

```
9207    AD0004    LDA    $0400
920A    0D0104    ORA    $0401
920D    F0F0      BEQ    $91FF
920F    207883    JSR    $8378
9212    F061      BEQ    $9275
9214    C985      CMP    #$85
;是 THEN 转 $921C
9216    F004      BEQ    $921C
9218    60        RTS
;转 GOTO 处理
9219    4C4891    JMP    $9148
;IF......THEN......的处理
921C    207583    JSR    $8375
921F    F0BE      BEQ    $91DF
9221    C90B      CMP    #$0B
9223    F0F4      BEQ    $9219
9225    60        RTS
    CLEAR 保留字入口
9226    D025      BNE    $924D
9228    A508      LDA    $08
922A    851E      STA    $1E
922C    A607      LDX    $07
922E    E8        INX
922F    D002      BNE    $9233
9231    E61E      INC    $1E
;程序区末址加一存入 $1D、$1E
9233    861D      STX    $1D
9235    A51E      LDA    $1E
9237    8520      STA    $20
9239    A61D      LDX    $1D
923B    E8        INX
923C    D002      BNE    $9240
923E    E620      INC    $20
;变量区首址加一存入 $1F、$20
9240    861F      STX    $1F
9242    A000      LDY    #$00
9244    98        TYA
9245    911D      STA    ($1D),Y
9247    C8        INY
9248    911D      STA    ($1D),Y
924A    4CFA98    JMP    $98FA
;进行 CLEAR 处理
924D    202892    JSR    $9228
9250    20829C    JSR    $9C82
9253    20EFA3    JSR    $A3EF
9256    AD0104    LDA    $0401
9259    C978      CMP    #$78
925B    B01B      BCS    $9278
925D    C520      CMP    $20
925F    9017      BCC    $9278
9261    D007      BNE    $926A
9263    AD0004    LDA    $0400
9266    C51F      CMP    $1F
9268    900E      BCC    $9278
926A    AD0004    LDA    $0400
926D    8503      STA    $03
926F    AD0104    LDA    $0401
;把用户设定的 RAM 自由区末址
存入 $03、$04 单元
9272    8504      STA    $04
9274    60        RTS
;出错处理,A=1,"SN"
9275    4C6D83    JMP    $836D
;出错处理,A=4,"IL"
9278    4C7083    JMP    $8370
;出错处理,A=H0B,"TM"
927B    4CE1A3    JMP    $A3E1
    FOR 保留字的入口
927E    F0F5      BEQ    $9275
;栈顶两单元出栈
9280    68        PLA
9281    855F      STA    $5F
9283    68        PLA
9284    8560      STA    $60
;下一行程序首址入栈
9286    A50D      LDA    $0D
9288    48        PHA
9289    A50E      LDA    $0E
928B    48        PHA
;当前程序行行号入栈
928C    A50B      LDA    $0B
928E    48        PHA
928F    A50C      LDA    $0C
9291    48        PHA
9292    20D48C    JSR    $8CD4
9295    20759F    JSR    $9F75
9298    A590      LDA    $90
929A    C902      CMP    #$02
```

```
929C    D0DD     BNE   $927B
;循环变量值的地址入栈
929E    A58C     LDA   $8C
92A0    48       PHA
92A1    A58D     LDA   $8D
92A3    48       PHA
92A4    20A285   JSR   $85A2
92A7    20829C   JSR   $9C82
;循环变量值首址出栈
92AA    68       PLA
92AB    858D     STA   $8D
92AD    68       PLA
92AE    858C     STA   $8C
;循环变量初值存入变量数值区
92B0    A000     LDY   #$00
92B2    AD0004   LDA   $0400
92B5    918C     STA   ($8C),Y
92B7    AD0104   LDA   $0401
92BA    C8       INY
92BB    918C     STA   ($8C),Y
;循环变量值首址入栈
92BD    A58C     LDA   $8C
92BF    48       PHA
92C0    A58D     LDA   $8D
92C2    48       PHA
92C3    20D48C   JSR   $8CD4
92C6    C988     CMP   #$88
92C8    D0AB     BNE   $9275
92CA    20A18C   JSR   $8CA1
92CD    20829C   JSR   $9C82
;循环变量终值入栈
92D0    AD0004   LDA   $0400
92D3    48       PHA
92D4    AD0104   LDA   $0401
92D7    48       PHA
92D8    A901     LDA   #$01
92DA    8D0004   STA   $0400
92DD    A900     LDA   #$00
92DF    8D0104   STA   $0401
92E2    207883   JSR   $8378
92E5    F00A     BEQ   $92F1
92E7    C989     CMP   #$89
92E9    D08A     BNE   $9275
;循环变量有步长的处理
92EB    20A18C   JSR   $8CA1
92EE    20829C   JSR   $9C82
;循环变量无步长的处理
92F1    AD0004   LDA   $0400
;步长入栈
92F4    48       PHA
92F5    AD0104   LDA   $0401
92F8    48       PHA
;程序区地址指针入栈
92F9    A509     LDA   $09
92FB    48       PHA
92FC    A50A     LDA   $0A
92FE    48       PHA
;循环数据栈标志入栈
92FF    A900     LDA   #$00
9301    48       PHA
9302    A9FD     LDA   #$FD
9304    48       PHA
;原栈顶标志入栈
9305    4C1C91   JMP   $911C
9308    A900     LDA   #$00
930A    4C8FA9   JMP   $A98F
        NEXT 保留字入口
;栈顶标志出栈
930D    68       PLA
930E    855F     STA   $5F
9310    68       PLA
9311    8560     STA   $60
;循环参数标志出栈
9313    68       PLA
9314    C9FD     CMP   #$FD
9316    D0F0     BNE   $9308
9318    68       PLA
9319    D0ED     BNE   $9308
;循环体首址出栈
931B    68       PLA
931C    8589     STA   $89
931E    68       PLA
931F    8588     STA   $88
;循环步长出栈
9321    68       PLA
9322    8583     STA   $83
9324    68       PLA
9325    8582     STA   $82
```

```
;循环终值出栈
9327    68       PLA
9328    8581     STA   $81
932A    68       PLA
932B    8580     STA   $80
;循环变量值首址出栈
932D    68       PLA
932E    857D     STA   $7D
9330    68       PLA
9331    857C     STA   $7C
;循环语句行行号出栈
9333    68       PLA
9334    8587     STA   $87
9336    68       PLA
9337    8586     STA   $86
;循环语句行的下一行首址出栈
9339    68       PLA
933A    8585     STA   $85
933C    68       PLA
933D    8584     STA   $84
933F    A000     LDY   #$00
9341    B17C     LDA   ($7C),Y
9343    8528     STA   $28
9345    C8       INY
9346    B17C     LDA   ($7C),Y
9348    8529     STA   $29
;令循环变量加一个步长
934A    208A93   JSR   $938A
934D    F002     BEQ   $9351
;循环变量小于、等于终值的处理
934F    9036     BCC   $9387
9351    A584     LDA   $84
9353    48       PHA
9354    850D     STA   $0D
9356    A585     LDA   $85
9358    48       PHA
9359    850E     STA   $0E
935B    A586     LDA   $86
935D    48       PHA
935E    850B     STA   $0B
9360    A587     LDA   $87
9362    48       PHA
9363    850C     STA   $0C
9365    A57C     LDA   $7C
9367    48       PHA
9368    A57D     LDA   $7D
936A    48       PHA
936B    A580     LDA   $80
936D    48       PHA
936E    A581     LDA   $81
9370    48       PHA
9371    A582     LDA   $82
9373    48       PHA
9374    A583     LDA   $83
9376    48       PHA
9377    A588     LDA   $88
9379    48       PHA
937A    8509     STA   $09
937C    A589     LDA   $89
937E    48       PHA
937F    850A     STA   $0A
9381    A900     LDA   #$00
9383    48       PHA
9384    A9FD     LDA   #$FD
9386    48       PHA
9387    4C1C91   JMP   $911C
;令循环变量增加一个步长,并判断是否达到终值
938A    A583     LDA   $83
938C    300C     BMI   $939A
938E    20A093   JSR   $93A0
9391    20038C   JSR   $8C03
9394    B002     BCS   $9398
9396    38       SEC
9397    60       RTS
9398    18       CLC
9399    60       RTS
939A    20A093   JSR   $93A0
939D    4C038C   JMP   $8C03
;令循环变量增加一个步长,存入循环变量区
93A0    A582     LDA   $82
93A2    852C     STA   $2C
93A4    A583     LDA   $83
93A6    852D     STA   $2D
93A8    20E88A   JSR   $8AE8
93AB    A000     LDY   #$00
93AD    A528     LDA   $28
93AF    917C     STA   ($7C),Y
93B1    C8       INY
```

```
93B2    A529      LDA    $29
93B4    917C      STA    ($7C),Y
93B6    A580      LDA    $80
93B8    852C      STA    $2C
93BA    A581      LDA    $81
93BC    852D      STA    $2D
93BE    60        RTS
93BF    4C6D83    JMP    $836D
```

INPUT 保留字的入口

```
93C2    F0FB      BEQ    $93BF
93C4    C922      CMP    #$22
93C6    D00B      BNE    $93D3
93C8    20DB94    JSR    $94DB
93CB    C92C      CMP    #$2C
93CD    F009      BEQ    $93D8
93CF    C93B      CMP    #$3B
93D1    D0EC      BNE    $93BF
93D3    A93F      LDA    #$3F
93D5    200DAC    JSR    $AC0D
```

;INPUT 后是“,”的处理

```
93D8    A549      LDA    $49
93DA    857E      STA    $7E
93DC    A548      LDA    $48
93DE    8580      STA    $80
93E0    2043B3    JSR    $B343
93E3    20D486    JSR    $86D4
93E6    A900      LDA    #$00
93E8    8552      STA    $52
93EA    A57E      LDA    $7E
93EC    C547      CMP    $47
93EE    F00B      BEQ    $93FB
93F0    C67E      DEC    $7E
```

;由$52单元记录本次输入的字符数

```
93F2    18        CLC
93F3    A91C      LDA    #$1C
93F5    6552      ADC    $52
93F7    8552      STA    $52
93F9    D0EF      BNE    $93EA
93FB    A580      LDA    $80
93FD    18        CLC
93FE    6552      ADC    $52
9400    8552      STA    $52
9402    20D48C    JSR    $8CD4
```

;判断A值,若是大写字母则建立变量区

```
9405    20759F    JSR    $9F75
9408    20F88F    JSR    $8FF8
940B    A552      LDA    $52
940D    8509      STA    $09
940F    A905      LDA    #$05
9411    850A      STA    $0A
9413    A590      LDA    $90
9415    290F      AND    #$0F
9417    C903      CMP    #$03
9419    F040      BEQ    $945B
```

;数型变量的处理

```
941B    20FEA4    JSR    $A4FE
941E    A509      LDA    $09
9420    8552      STA    $52
9422    A000      LDY    #$00
9424    A528      LDA    $28
9426    918C      STA    ($8C),Y
9428    C8        INY
9429    A529      LDA    $29
942B    918C      STA    ($8C),Y
942D    200190    JSR    $9001
9430    20D48C    JSR    $8CD4
9433    C92C      CMP    #$2C
9435    D087      BNE    $93BE
9437    20A18C    JSR    $8CA1
943A    20F88F    JSR    $8FF8
943D    A552      LDA    $52
943F    8509      STA    $09
9441    A905      LDA    #$05
9443    850A      STA    $0A
9445    A000      LDY    #$00
9447    B109      LDA    ($09),Y
9449    C92C      CMP    #$2C
944B    D008      BNE    $9455
944D    E652      INC    $52
944F    200190    JSR    $9001
9452    4C0294    JMP    $9402
9455    200190    JSR    $9001
9458    4CD393    JMP    $93D3
```

;串型变量的处理

```
945B    20D48C    JSR    $8CD4
945E    A900      LDA    #$00
9460    851B      STA    $1B
9462    A000      LDY    #$00
```

```
9464   A200     LDX   #$00
9466   B109     LDA   ($09),Y
9468   F010     BEQ   $947A
946A   C92C     CMP   #$2C
946C   F00C     BEQ   $947A
946E   9D0007   STA   $0700,X
9471   E61B     INC   $1B
9473   E8       INX
9474   20A18C   JSR   $8CA1
9477   4C6694   JMP   $9466
947A   A51B     LDA   $1B
947C   8D0004   STA   $0400
947F   A900     LDA   #$00
9481   8D0104   STA   $0401
9484   A907     LDA   #$07
9486   8D0204   STA   $0402
9489   20EF82   JSR   $82EF
948C   A509     LDA   $09
948E   8552     STA   $52
9490   4C2D94   JMP   $942D
9493   4C6D83   JMP   $836D
```

SWAP 保留字的入口

```
9496   F0FB     BEQ   $9493
9498   20759F   JSR   $9F75
```

;第一变量类型入栈

```
949B   A590     LDA   $90
949D   290F     AND   #$0F
949F   48       PHA
```

;第一变量值入栈

```
94A0   A58C     LDA   $8C
94A2   48       PHA
94A3   A58D     LDA   $8D
94A5   48       PHA
94A6   2094A5   JSR   $A594
94A9   20D48C   JSR   $8CD4
```

;第二变量类型存入 $90

第二变量首址存入 $8C、$8D

```
94AC   20759F   JSR   $9F75
94AF   68       PLA
94B0   851A     STA   $1A
94B2   68       PLA
94B3   8519     STA   $19
94B5   A590     LDA   $90
94B7   290F     AND   #$0F
94B9   8590     STA   $90
94BB   68       PLA
94BC   C590     CMP   $90
94BE   D018     BNE   $94D8
94C0   AA       TAX
94C1   C902     CMP   #$02
94C3   F002     BEQ   $94C7
94C5   A221     LDX   #$21
94C7   A000     LDY   #$00
94C9   B18C     LDA   ($8C),Y
94CB   48       PHA
94CC   B119     LDA   ($19),Y
94CE   918C     STA   ($8C),Y
94D0   68       PLA
94D1   9119     STA   ($19),Y
94D3   C8       INY
94D4   CA       DEX
94D5   D0F2     BNE   $94C9
94D7   60       RTS
94D8   4CE1A3   JMP   $A3E1
```

INPUT 后有引号时,把引号中的内容读出送入七页显示

```
94DB   20A18C   JSR   $8CA1
94DE   A000     LDY   #$00
94E0   B109     LDA   ($09),Y
94E2   F0AF     BEQ   $9493
94E4   C922     CMP   #$22
94E6   F006     BEQ   $94EE
94E8   200DAC   JSR   $AC0D
94EB   4CDB94   JMP   $94DB
```

INPUT 中读到右引号时的处理

```
94EE   207583   JSR   $8375
94F1   F0A0     BEQ   $9493
94F3   4CA18C   JMP   $8CA1
```

LINPUT 保留字入口

```
94F6   F09B     BEQ   $9493
94F8   C922     CMP   #$22
94FA   D00B     BNE   $9507
```

;LINPUT 后有引号的处理

```
94FC   20DB94   JSR   $94DB
94FF   C92C     CMP   #$2C
9501   F004     BEQ   $9507
9503   C93B     CMP   #$3B
9505   D08C     BNE   $9493
```

```
;LINPUT 提示后的处理
9507    2043B3    JSR    $B343
950A    20D486    JSR    $86D4
950D    20D48C    JSR    $8CD4
9510    20759F    JSR    $9F75
9513    A590      LDA    $90
9515    290F      AND    #$0F
9517    C903      CMP    #$03
9519    D0BD      BNE    $94D8
951B    A200      LDX    #$00
951D    A000      LDY    #$00
951F    BD0005    LDA    $0500,X
9522    C920      CMP    #$20
9524    D007      BNE    $952D
9526    E8        INX
9527    BD0005    LDA    $0500,X
952A    F00E      BEQ    $953A
952C    CA        DEX
952D    BD0005    LDA    $0500,X
9530    F008      BEQ    $953A
9532    9D0007    STA    $0700,X
9535    E8        INX
9536    C8        INY
9537    4C2D95    JMP    $952D
953A    8C0004    STY    $0400
953D    A900      LDA    #$00
953F    8D0104    STA    $0401
9542    A905      LDA    #$05
9544    8D0204    STA    $0402
9547    4CEF82    JMP    $82EF
954A    4C6D83    JMP    $836D
   ON 保留字的入口
954D    206D9C    JSR    $9C6D
9550    20D48C    JSR    $8CD4
9553    48        PHA
9554    20A18C    JSR    $8CA1
9557    AE0004    LDX    $0400
955A    F034      BEQ    $9590
955C    207883    JSR    $8378
955F    F032      BEQ    $9593
9561    C90B      CMP    #$0B
9563    D0E5      BNE    $954A
9565    CA        DEX
9566    F00E      BEQ    $9576
9568    202799    JSR    $9927
956B    207883    JSR    $8378
956E    F023      BEQ    $9593
9570    2094A5    JSR    $A594
9573    4C5C95    JMP    $955C
   开关参数为 1 的处理
9576    68        PLA
9577    38        SEC
9578    E980      SBC    #$80
957A    C905      CMP    #$05
957C    B0CC      BCS    $954A
957E    0A        ASL
957F    AA        TAX
9580    BDC6C2    LDA    $C2C6,X
9583    8519      STA    $19
9585    BDC7C2    LDA    $C2C7,X
9588    851A      STA    $1A
958A    20D48C    JSR    $8CD4
958D    6C1900    JMP    ($0019)
9590    209895    JSR    $9598
9593    68        PLA
9594    60        RTS
9595    202799    JSR    $9927
9598    207883    JSR    $8378
959B    F0F7      BEQ    $9594
959D    C90B      CMP    #$0B
959F    F0F4      BEQ    $9595
95A1    2094A5    JSR    $A594
95A4    4C9895    JMP    $9598
95A7    4C7083    JMP    $8370
   KEY 保留字的入口
95AA    C986      CMP    #$86
;是 KEYLIST 转$95F3
95AC    F045      BEQ    $95F3
;以下是 KEY 指令的处理
95AE    206D9C    JSR    $9C6D
95B1    F0F4      BEQ    $95A7
95B3    48        PHA
95B4    2094A5    JSR    $A594
95B7    207C9C    JSR    $9C7C
95BA    68        PLA
95BB    38        SEC
95BC    E901      SBC    #$01
95BE    C909      CMP    #$09
```

```
95C0    B0E5      BCS    $95A7
95C2    0A        ASL
95C3    0A        ASL
95C4    0A        ASL
95C5    0A        ASL
95C6    AA        TAX
95C7    AD0004    LDA    $0400
95CA    F021      BEQ    $95ED
95CC    A000      LDY    #$00
95CE    AD0104    LDA    $0401
95D1    8519      STA    $19
95D3    AD0204    LDA    $0402
95D6    851A      STA    $1A
95D8    A90F      LDA    #$0F
95DA    8590      STA    $90
95DC    B119      LDA    ($19),Y
95DE    9D8004    STA    $0480,X
95E1    E8        INX
95E2    C8        INY
95E3    CE0004    DEC    $0400
95E6    F005      BEQ    $95ED
95E8    C690      DEC    $90
95EA    D0F0      BNE    $95DC
95EC    60        RTS
95ED    A900      LDA    #$00
95EF    9D8004    STA    $0480,X
95F2    60        RTS
    KEYLIST 保留字入口
95F3    20A18C    JSR    $8CA1
95F6    A930      LDA    #$30
95F8    8590      STA    $90
95FA    A900      LDA    #$00
95FC    8574      STA    $74
95FE    A946      LDA    #$46
9600    20F087    JSR    $87F0
9603    E690      INC    $90
9605    A590      LDA    $90
;送功能键号码 1～8
9607    20F087    JSR    $87F0
960A    A920      LDA    #$20
;送一个空格
960C    20F087    JSR    $87F0
960F    A574      LDA    $74
9611    0A        ASL
9612    0A        ASL
9613    0A        ASL
9614    0A        ASL
9615    A8        TAY
9616    B98004    LDA    $0480,Y
9619    F021      BEQ    $963C
961B    C920      CMP    #$20
961D    9007      BCC    $9626
961F    20F087    JSR    $87F0
9622    C8        INY
9623    4C1696    JMP    $9616
;读到定义字符小于 H20 的处理
9626    48        PHA
9627    A928      LDA    #$28
;送“(”
9629    20F087    JSR    $87F0
962C    68        PLA
962D    18        CLC
962E    6940      ADC    #$40
;送“M”
9630    20F087    JSR    $87F0
9633    A929      LDA    #$29
;送“)”
9635    20F087    JSR    $87F0
9638    C8        INY
9639    4C1696    JMP    $9616
963C    20FB87    JSR    $87FB
963F    2007AC    JSR    $AC07
9642    E674      INC    $74
9644    A574      LDA    $74
9646    C908      CMP    #$08
9648    90B4      BCC    $95FE
964A    60        RTS
964B    4C6D83    JMP    $836D
    DIM 保留字入口
964E    20C3A1    JSR    $A1C3
9651    B0F8      BCS    $964B
9653    857C      STA    $7C
9655    A902      LDA    #$02
;预置变量类型
9657    8584      STA    $84
9659    A900      LDA    #$00
;预置变量名的第二个字母
965B    857D      STA    $7D
```

```
965D    20AD8C    JSR    $8CAD
9660    20BBA1    JSR    $A1BB
9663    B005      BCS    $966A
9665    857D      STA    $7D
9667    20A18C    JSR    $8CA1
966A    20D48C    JSR    $8CD4
966D    C924      CMP    #$24
966F    D005      BNE    $9676
9671    E684      INC    $84
9673    207583    JSR    $8375
9676    C928      CMP    #$28
9678    D0D1      BNE    $964B
967A    A900      LDA    #$00
;预置数组维数标志
967C    8582      STA    $82
967E    20649C    JSR    $9C64
;存第一下标
9681    857E      STA    $7E
9683    20D48C    JSR    $8CD4
9686    C929      CMP    #$29
9688    F00E      BEQ    $9698
968A    E682      INC    $82
968C    206A9C    JSR    $9C6A
;存第二下标
968F    857F      STA    $7F
9691    207883    JSR    $8378
9694    C929      CMP    #$29
9696    D0B3      BNE    $964B
9698    20A18C    JSR    $8CA1
969B    20B296    JSR    $96B2
969E    207883    JSR    $8378
96A1    F009      BEQ    $96AC
96A3    2094A5    JSR    $A594
96A6    20D48C    JSR    $8CD4
;定义后面的数组
96A9    4C4E96    JMP    $964E
96AC    60        RTS
96AD    A909      LDA    #$09
;出错处理,代码=DD
96AF    4C8FA9    JMP    $A98F
;以下为每个下标变量分配变量区
96B2    207897    JSR    $9778
96B5    B0F6      BCS    $96AD
96B7    A901      LDA    #$01
96B9    852C      STA    $2C
96BB    20CDA1    JSR    $A1CD
96BE    A57E      LDA    $7E
96C0    8528      STA    $28
96C2    E628      INC    $28
96C4    D002      BNE    $96C8
96C6    E629      INC    $29
96C8    A582      LDA    $82
96CA    F00A      BEQ    $96D6
96CC    A57F      LDA    $7F
96CE    852C      STA    $2C
96D0    E62C      INC    $2C
96D2    D002      BNE    $96D6
96D4    E62D      INC    $2D
96D6    205C8B    JSR    $8B5C
96D9    2003A1    JSR    $A103
96DC    852C      STA    $2C
96DE    A900      LDA    #$00
96E0    852D      STA    $2D
96E2    205C8B    JSR    $8B5C
96E5    A905      LDA    #$05
96E7    852C      STA    $2C
96E9    A582      LDA    $82
96EB    F002      BEQ    $96EF
96ED    E62C      INC    $2C
96EF    20398B    JSR    $8B39
96F2    B01E      BCS    $9712
96F4    A58C      LDA    $8C
96F6    8519      STA    $19
96F8    852C      STA    $2C
96FA    A58D      LDA    $8D
96FC    851A      STA    $1A
96FE    852D      STA    $2D
9700    20398B    JSR    $8B39
9703    B00D      BCS    $9712
9705    A504      LDA    $04
9707    852D      STA    $2D
9709    A503      LDA    $03
970B    852C      STA    $2C
970D    20F78B    JSR    $8BF7
9710    9003      BCC    $9715
9712    4C568A    JMP    $8A56
;以下把定义数组的有关内容写入数组变量区
9715    A000      LDY    #$00
```

```
9717    A590      LDA    $90
;存类型
9719    918C      STA    ($8C),Y
971B    C8        INY
971C    A58A      LDA    $8A
971E    918C      STA    ($8C),Y
9720    C8        INY
9721    A58B      LDA    $8B
;存数组名
9723    918C      STA    ($8C),Y
9725    C8        INY
9726    A582      LDA    $82
;存维数标志
9728    918C      STA    ($8C),Y
972A    C8        INY
972B    A57E      LDA    $7E
;存第一下标
972D    918C      STA    ($8C),Y
972F    A582      LDA    $82
9731    F005      BEQ    $9738
9733    C8        INY
9734    A57F      LDA    $7F
;存第二下标
9736    918C      STA    ($8C),Y
9738    A528      LDA    $28
973A    851F      STA    $1F
973C    A529      LDA    $29
973E    8520      STA    $20
9740    18        CLC
9741    A205      LDX    #$05
9743    A582      LDA    $82
9745    F001      BEQ    $9748
9747    E8        INX
9748    8A        TXA
9749    6519      ADC    $19
974B    8519      STA    $19
974D    A900      LDA    #$00
974F    651A      ADC    $1A
9751    851A      STA    $1A
9753    A000      LDY    #$00
9755    A900      LDA    #$00
;把本数组值的存储单元全部置0
9757    9119      STA    ($19),Y
9759    E619      INC    $19
975B    D002      BNE    $975F
975D    E61A      INC    $1A
975F    A51A      LDA    $1A
9761    C520      CMP    $20
9763    D0EE      BNE    $9753
9765    A519      LDA    $19
9767    C51F      CMP    $1F
9769    D0E8      BNE    $9753
976B    A000      LDY    #$00
976D    A900      LDA    #$00
;变量区的最后置三个零
976F    9119      STA    ($19),Y
9771    C8        INY
9772    9119      STA    ($19),Y
9774    C8        INY
9775    9119      STA    ($19),Y
9777    60        RTS
;查出变量区末址存入$8C、$8D
9778    A57C      LDA    $7C
977A    858A      STA    $8A
977C    A57D      LDA    $7D
977E    858B      STA    $8B
9780    A584      LDA    $84
9782    0980      ORA    #$80
;数组类型标志加H80后送入$90
9784    8590      STA    $90
;检索变量区,检索到令C=1返回;否则C=0
9786    200EA1    JSR    $A10E
;把变量区末址存入$8C、$8D
9789    4C07A0    JMP    $A007
    SAVE 保留字入口
;检索变量名
978C    206998    JSR    $9869
978F    20B498    JSR    $98B4
9792    A902      LDA    #$02
9794    8D0005    STA    $0500
9797    A505      LDA    $05
9799    8D1405    STA    $0514
979C    852C      STA    $2C
979E    A506      LDA    $06
;程序区首址存入$514、$515
和$2C、$2D单元
97A0    8D1505    STA    $0515
97A3    852D      STA    $2D
```

```
97A5    A507      LDA    $07
97A7    8528      STA    $28
97A9    A508      LDA    $08
;程序区末址存入$28、$29单元
97AB    8529      STA    $29
97AD    206F8A    JSR    $8A6F
97B0    A528      LDA    $28
97B2    8D1205    STA    $0512
97B5    A529      LDA    $29
;程序量存入$512、$513单元
97B7    8D1305    STA    $0513
;送显"WRITING"
97BA    20C397    JSR    $97C3
;输出数据至磁带
97BD    20FCB4    JSR    $B4FC
97C0    4C0DB5    JMP    $B50D
    "LOAD、SAVE"中各种提示的送显
97C3    A946      LDA    #$46
97C5    8552      STA    $52
97C7    A9AA      LDA    #$AA
97C9    8553      STA    $53
;送显"WRITING"
97CB    20F887    JSR    $87F8
97CE    F018      BEQ    $97E8
97D0    A94F      LDA    #$4F
97D2    8552      STA    $52
97D4    A9AA      LDA    #$AA
97D6    8553      STA    $53
;送显"LOADING"
97D8    20F887    JSR    $87F8
97DB    F00B      BEQ    $97E8
97DD    A958      LDA    #$58
97DF    8552      STA    $52
97E1    A9AA      LDA    #$AA
97E3    8553      STA    $53
;送显"SKIP"
97E5    20F887    JSR    $87F8
97E8    A901      LDA    #$01
97EA    8552      STA    $52
97EC    A905      LDA    #$05
97EE    8553      STA    $53
;送显存入$501以后的文件名
97F0    20F887    JSR    $87F8
97F3    4C07AC    JMP    $AC07
97F6    20DD97    JSR    $97DD
;从磁带中读出文件名存入$501以后
97F9    4C0C98    JMP    $980C
    LOAD保留字入口
97FC    A200      LDX    #$00
97FE    869F      STX    $9F
9800    C98B      CMP    #$8B
9802    D005      BNE    $9809
9804    C69F      DEC    $9F
9806    20A18C    JSR    $8CA1
;把存入$780以后的文件名取出存入$580以后
9809    206998    JSR    $9869
;读出文件名存入$501以后
980C    2030B4    JSR    $B430
980F    B053      BCS    $9864
;核对文件名
9811    20C898    JSR    $98C8
9814    B0E0      BCS    $97F6
;文件名相同的处理
9816    AD0005    LDA    $0500
9819    C902      CMP    #$02
981B    D0D9      BNE    $97F6
;送显"LOADING"文件名
981D    20D097    JSR    $97D0
9820    A59F      LDA    $9F
9822    F006      BEQ    $982A
;LOAD、PRINT的处理
9824    20F9B3    JSR    $B3F9
9827    B03B      BCS    $9864
9829    60        RTS
;LOAD的处理
982A    A505      LDA    $05
982C    8D1405    STA    $0514
982F    A506      LDA    $06
9831    8D1505    STA    $0515
;从磁带中读出程序,若有错则令C=1
9834    2001B4    JSR    $B401
9837    B02B      BCS    $9864
9839    A5AD      LDA    $AD
983B    8507      STA    $07
983D    A5AE      LDA    $AE
983F    8508      STA    $08
9841    205D98    JSR    $985D
9844    20ABB3    JSR    $B3AB
```

```
9847   A5AD     LDA   $AD
9849   851D     STA   $1D
984B   A5AE     LDA   $AE
984D   851E     STA   $1E
984F   205D98   JSR   $985D
9852   20ABB3   JSR   $B3AB
9855   A5AD     LDA   $AD
9857   851F     STA   $1F
9859   A5AE     LDA   $AE
985B   8520     STA   $20
985D   A000     LDY   #$00
985F   A900     LDA   #$00
9861   91AD     STA   ($AD),Y
9863   60       RTS
9864   A911     LDA   #$11
9866   4C8FA9   JMP   $A98F
    执行SAVE、LOAD时用于把存在$780
    以后的文件名，送入$580开始的以后单元
9869   20A198   JSR   $98A1
986C   A900     LDA   #$00
986E   85B5     STA   $B5
9870   207883   JSR   $8378
9873   F02B     BEQ   $98A0
9875   207C9C   JSR   $9C7C
;把存在$401、$402指定地址
中的数据送入$580
9878   AD0104   LDA   $0401
987B   8519     STA   $19
987D   AD0204   LDA   $0402
9880   851A     STA   $1A
9882   AD0004   LDA   $0400
9885   F019     BEQ   $98A0
9887   A000     LDY   #$00
9889   A200     LDX   #$00
988B   B119     LDA   ($19),Y
988D   9D8005   STA   $0580,X
9890   E8       INX
9891   C8       INY
9892   CE0004   DEC   $0400
9895   D0F4     BNE   $988B
9897   A900     LDA   #$00
9899   9D8005   STA   $0580,X
989C   A901     LDA   #$01
989E   85B5     STA   $B5
```

```
98A0   60       RTS
;令$580～$58F=H20、$590=0
98A1   A200     LDX   #$00
98A3   A010     LDY   #$10
98A5   A920     LDA   #$20
98A7   9D8005   STA   $0580,X
98AA   E8       INX
98AB   88       DEY
98AC   D0F9     BNE   $98A7
98AE   A900     LDA   #$00
98B0   9D8005   STA   $0580,X
98B3   60       RTS
;把$580开始的以后单元的数据
送入$501开始的各单元
98B4   A200     LDX   #$00
98B6   A010     LDY   #$10
98B8   BD8005   LDA   $0580,X
98BB   9D0105   STA   $0501,X
98BE   E8       INX
98BF   88       DEY
98C0   D0F6     BNE   $98B8
98C2   A900     LDA   #$00
98C4   9D0105   STA   $0501,X
98C7   60       RTS
;执行LOAD时，核对欲读的文件名是否与已
读到的文件名相同
98C8   A5B5     LDA   $B5
98CA   F010     BEQ   $98DC
98CC   A200     LDX   #$00
98CE   A010     LDY   #$10
98D0   BD0105   LDA   $0501,X
98D3   DD8005   CMP   $0580,X
98D6   D006     BNE   $98DE
98D8   E8       INX
98D9   88       DEY
98DA   D0F4     BNE   $98D0
98DC   18       CLC
98DD   60       RTS
98DE   38       SEC
98DF   60       RTS
    RESTORE保留字入口
98E0   F018     BEQ   $98FA
;检索RESTORE后的目标行号
98E2   207291   JSR   $9172
```

```
98E5    209895    JSR    $9598
98E8    A903      LDA    #$03
98EA    18        CLC
98EB    655B      ADC    $5B
98ED    85B2      STA    $B2
98EF    A900      LDA    #$00
98F1    655C      ADC    $5C
98F3    85B3      STA    $B3
98F5    A900      LDA    #$00
98F7    85B4      STA    $B4
98F9    60        RTS
;RESTORE 后无行号的处理
98FA    A505      LDA    $05
98FC    18        CLC
98FD    6903      ADC    #$03
98FF    85B2      STA    $B2
9901    A900      LDA    #$00
9903    6506      ADC    $06
9905    4CF398    JMP    $98F3
     执行 CONT 的处理
9908    20C78C    JSR    $8CC7
990B    C980      CMP    #$80
990D    B015      BCS    $9924
990F    C912      CMP    #$12
9911    F014      BEQ    $9927
9913    C911      CMP    #$11
9915    F010      BEQ    $9927
9917    C90B      CMP    #$0B
9919    F00C      BEQ    $9927
991B    C922      CMP    #$22
991D    F015      BEQ    $9934
991F    20C3A1    JSR    $A1C3
9922    901E      BCC    $9942
9924    4CA18C    JMP    $8CA1
9927    20A18C    JSR    $8CA1
992A    20A18C    JSR    $8CA1
992D    4CA18C    JMP    $8CA1
9930    68        PLA
9931    4CA18C    JMP    $8CA1
9934    48        PHA
9935    20AD8C    JSR    $8CAD
9938    C922      CMP    #$22
993A    F0F4      BEQ    $9930
993C    29FF      AND    #$FF
993E    D0F5      BNE    $9935
9940    68        PLA
9941    60        RTS
9942    48        PHA
9943    20AD8C    JSR    $8CAD
9946    20BBA1    JSR    $A1BB
9949    90F8      BCC    $9943
994B    68        PLA
994C    60        RTS
     READ 保留字的入口
994D    20759F    JSR    $9F75
9950    20F88F    JSR    $8FF8
;检查第一个 DATA 语句
9953    A5B2      LDA    $B2
9955    8509      STA    $09
9957    A5B3      LDA    $B3
9959    850A      STA    $0A
995B    A5B4      LDA    $B4
995D    D026      BNE    $9985
995F    20C78C    JSR    $8CC7
9962    F00A      BEQ    $996E
9964    C991      CMP    #$91
9966    F016      BEQ    $997E
9968    200899    JSR    $9908
996B    4C5F99    JMP    $995F
996E    20AD8C    JSR    $8CAD
9971    F006      BEQ    $9979
9973    202799    JSR    $9927
9976    4C5F99    JMP    $995F
9979    A903      LDA    #$03
997B    4C8FA9    JMP    $A98F
;读到 DATA 语句后的处理
997E    A9FF      LDA    #$FF
9980    85B4      STA    $B4
9982    200899    JSR    $9908
9985    A590      LDA    $90
9987    290F      AND    #$0F
9989    C903      CMP    #$03
998B    F03E      BEQ    $99CB
;从 DATA 语句中读数值的处理
998D    20FEA4    JSR    $A4FE
9990    A000      LDY    #$00
9992    A528      LDA    $28
9994    918C      STA    ($8C),Y
```

```
9996    A529     LDA    $29
9998    C8       INY
9999    918C     STA    ($8C),Y
999B    207883   JSR    $8378
999E    F006     BEQ    $99A6
99A0    2094A5   JSR    $A594
99A3    4CAA99   JMP    $99AA
99A6    A900     LDA    #$00
99A8    85B4     STA    $B4
99AA    A509     LDA    $09
99AC    85B2     STA    $B2
99AE    A50A     LDA    $0A
99B0    85B3     STA    $B3
99B2    200190   JSR    $9001
99B5    20D48C   JSR    $8CD4
99B8    C92C     CMP    #$2C
99BA    F001     BEQ    $99BD
99BC    60       RTS
99BD    20A18C   JSR    $8CA1
99C0    20D48C   JSR    $8CD4
99C3    20C3A1   JSR    $A1C3
99C6    B0F4     BCS    $99BC
99C8    4C4D99   JMP    $994D
;从DATA语句中读字串的处理
99CB    A900     LDA    #$00
99CD    851B     STA    $1B
99CF    AA       TAX
99D0    A92C     LDA    #$2C
99D2    853C     STA    $3C
99D4    20D48C   JSR    $8CD4
99D7    C922     CMP    #$22
99D9    D005     BNE    $99E0
99DB    853C     STA    $3C
99DD    20A18C   JSR    $8CA1
99E0    B109     LDA    ($09),Y
99E2    F020     BEQ    $9A04
99E4    C53C     CMP    $3C
99E6    F00C     BEQ    $99F4
99E8    9D0007   STA    $0700,X
99EB    E61B     INC    $1B
99ED    E8       INX
99EE    20A18C   JSR    $8CA1
99F1    4CE099   JMP    $99E0
99F4    20A18C   JSR    $8CA1
99F7    20D48C   JSR    $8CD4
99FA    C92C     CMP    #$2C
99FC    D00A     BNE    $9A08
99FE    20A18C   JSR    $8CA1
9A01    4C089A   JMP    $9A08
9A04    A900     LDA    #$00
9A06    85B4     STA    $B4
9A08    A51B     LDA    $1B
9A0A    8D0004   STA    $0400
9A0D    A900     LDA    #$00
9A0F    8D0104   STA    $0401
9A12    A907     LDA    #$07
9A14    8D0204   STA    $0402
9A17    20EF82   JSR    $82EF
9A1A    4CAA99   JMP    $99AA
        PLAY保留字的入口
9A1D    207C9C   JSR    $9C7C
9A20    AD0004   LDA    $0400
9A23    F060     BEQ    $9A85
9A25    A509     LDA    $09
9A27    48       PHA
9A28    A50A     LDA    $0A
9A2A    48       PHA
9A2B    AD0104   LDA    $0401
9A2E    8509     STA    $09
;取系统A的弦乐资料存于$700～$720
$700～$720单元
9A30    AD0204   LDA    $0402
9A33    850A     STA    $0A
9A35    A907     LDA    #$07
9A37    8592     STA    $92
9A39    A900     LDA    #$00
9A3B    8591     STA    $91
9A3D    85C6     STA    $C6
9A3F    20409C   JSR    $9C40
;取系统B的弦乐资料存入$721～$741
$721～$741单元
9A42    A921     LDA    #$21
9A44    8591     STA    $91
9A46    85C7     STA    $C7
9A48    20409C   JSR    $9C40
;取系统C的弦乐资料存入$742～$762
$742～$762单元
9A4B    A942     LDA    #$42
```

```
9A4D    8591      STA    $91
9A4F    85C8      STA    $C8
9A51    20409C    JSR    $9C40
9A54    A901      LDA    #$01
9A56    85CB      STA    $CB
9A58    85CC      STA    $CC
9A5A    85CD      STA    $CD
9A5C    A97F      LDA    #$7F
9A5E    8D0140    STA    $4001
9A61    8D0540    STA    $4005
9A64    A907      LDA    #$07
9A66    85C9      STA    $C9
9A68    8D1540    STA    $4015
9A6B    A90A      LDA    #$0A
9A6D    8563      STA    $63
9A6F    20D486    JSR    $86D4
9A72    A5C9      LDA    $C9
9A74    3010      BMI    $9A86
9A76    D0F7      BNE    $9A6F
9A78    A900      LDA    #$00
9A7A    8563      STA    $63
9A7C    8D1540    STA    $4015
9A7F    68        PLA
9A80    850A      STA    $0A
9A82    68        PLA
9A83    8509      STA    $09
9A85    60        RTS
9A86    A900      LDA    #$00
9A88    8563      STA    $63
9A8A    8D1540    STA    $4015
9A8D    4C6D83    JMP    $836D
```

中断中执行 PLAY 语句的主控程序

```
9A90    8A        TXA
9A91    48        PHA
9A92    98        TYA
9A93    48        PHA
9A94    A9C0      LDA    #$C0
9A96    8D1740    STA    $4017
9A99    20A29A    JSR    $9AA2
9A9C    68        PLA
9A9D    A8        TAY
9A9E    68        PLA
9A9F    AA        TAX
9AA0    68        PLA
9AA1    40        RTI
```

执行 PLAY 语句的辅助程序

```
9AA2    A902      LDA    #$02
9AA4    85CA      STA    $CA
9AA6    A904      LDA    #$04
9AA8    85CE      STA    $CE
9AAA    A5C9      LDA    $C9
9AAC    3009      BMI    $9AB7
9AAE    20B89A    JSR    $9AB8
9AB1    46CE      LSR    $CE
9AB3    C6CA      DEC    $CA
9AB5    10F3      BPL    $9AAA
9AB7    60        RTS
```

读弦乐资料子程序

```
9AB8    A5C9      LDA    $C9
9ABA    25CE      AND    $CE
9ABC    F0F9      BEQ    $9AB7
9ABE    A6CA      LDX    $CA
9AC0    D6CB      DEC    $CB,X
9AC2    D0F3      BNE    $9AB7
9AC4    B5C6      LDA    $C6,X
9AC6    8591      STA    $91
9AC8    A900      LDA    #$00
9ACA    85C2      STA    $C2
9ACC    A000      LDY    #$00
9ACE    B191      LDA    ($91),Y
9AD0    D011      BNE    $9AE3
9AD2    A5C9      LDA    $C9
9AD4    38        SEC
9AD5    E5CE      SBC    $CE
9AD7    85C9      STA    $C9
9AD9    8D1540    STA    $4015
9ADC    A6CA      LDX    $CA
9ADE    A591      LDA    $91
9AE0    95C6      STA    $C6,X
9AE2    60        RTS
```

输入弦乐资料的处理

```
9AE3    E691      INC    $91
9AE5    C920      CMP    #$20
9AE7    F0E3      BEQ    $9ACC
9AE9    C94D      CMP    #$4D
9AEB    D018      BNE    $9B05
;读到"M"时
9AED    B191      LDA    ($91),Y
```

```
9AEF    F079    BEQ   $9B6A
9AF1    A230    LDX   #$30
9AF3    C930    CMP   #$30
9AF5    F006    BEQ   $9AFD
9AF7    C931    CMP   #$31
9AF9    D06F    BNE   $9B6A
9AFB    A200    LDX   #$00
9AFD    E691    INC   $91
9AFF    A4CA    LDY   $CA
9B01    96CF    STX   $CF,Y
9B03    10C7    BPL   $9ACC
9B05    C94F    CMP   #$4F
9B07    D015    BNE   $9B1E
;读到“O”时
9B09    B191    LDA   ($91),Y
9B0B    F05D    BEQ   $9B6A
9B0D    E691    INC   $91
9B0F    C936    CMP   #$36
9B11    B057    BCS   $9B6A
9B13    38      SEC
9B14    E930    SBC   #$30
9B16    9052    BCC   $9B6A
9B18    A6CA    LDX   $CA
9B1A    95BF    STA   $BF,X
9B1C    10AE    BPL   $9ACC
9B1E    C954    CMP   #$54
9B20    D017    BNE   $9B39
;读到“T”时
9B22    B191    LDA   ($91),Y
9B24    F044    BEQ   $9B6A
9B26    E691    INC   $91
9B28    C939    CMP   #$39
9B2A    B03E    BCS   $9B6A
9B2C    38      SEC
9B2D    E930    SBC   #$30
9B2F    F039    BEQ   $9B6A
9B31    9037    BCC   $9B6A
9B33    A6CA    LDX   $CA
9B35    95B9    STA   $B9,X
9B37    1093    BPL   $9ACC
9B39    C956    CMP   #$56
9B3B    D032    BNE   $9B6F
;读到“V”时
9B3D    B191    LDA   ($91),Y
```

```
9B3F    F029    BEQ   $9B6A
9B41    E691    INC   $91
9B43    C93A    CMP   #$3A
9B45    B023    BCS   $9B6A
9B47    38      SEC
9B48    E930    SBC   #$30
9B4A    901E    BCC   $9B6A
9B4C    C901    CMP   #$01
9B4E    D014    BNE   $9B64
9B50    B191    LDA   ($91,Y)
9B52    C936    CMP   #$36
9B54    B00C    BCS   $9B62
9B56    38      SEC
9B57    E930    SBC   #$30
9B59    9007    BCC   $9B62
9B5B    18      CLC
9B5C    690A    ADC   #$0A
9B5E    E691    INC   $91
9B60    D002    BNE   $9B64
9B62    A901    LDA   #$01
9B64    A6CA    LDX   $CA
9B66    95BC    STA   $BC,X
9B68    10CD    BPL   $9B37
9B6A    A980    LDA   #$80
9B6C    85C9    STA   $C9
9B6E    60      RTS
9B6F    C959    CMP   #$59
9B71    D016    BNE   $9B89
;读到“Y”时
9B73    B191    LDA   ($91),Y
9B75    C934    CMP   #$34
9B77    B0F1    BCS   $9B6A
9B79    38      SEC
9B7A    E930    SBC   #$30
9B7C    90EC    BCC   $9B6A
9B7E    E691    INC   $91
9B80    4A      LSR
9B81    6A      ROR
9B82    6A      ROR
9B83    A6CA    LDX   $CA
9B85    95D2    STA   $D2,X
9B87    10AE    BPL   $9B37
9B89    C923    CMP   #$23
9B8B    D006    BNE   $9B93
```

```
;读到“#”时
9B8D    A902      LDA    #$02
9B8F    85C2      STA    $C2
9B91    10A4      BPL    $9B37
9B93    C952      CMP    #$52
9B95    D014      BNE    $9BAB
;读到“R”时
9B97    200E9C    JSR    $9C0E
9B9A    A5CA      LDA    $CA
9B9C    0A        ASL
9B9D    0A        ASL
9B9E    A8        TAY
9B9F    A990      LDA    #$90
9BA1    C008      CPY    #$08
9BA3    D002      BNE    $9BA7
9BA5    A900      LDA    #$00
9BA7    990040    STA    $4000,Y
9BAA    60        RTS
9BAB    C941      CMP    #$41
9BAD    90BB      BCC    $9B6A
9BAF    C948      CMP    #$48
9BB1    B0B7      BCS    $9B6A
;音名为 A~G 的处理
9BB3    48        PHA
9BB4    200E9C    JSR    $9C0E
9BB7    A6CA      LDX    $CA
9BB9    8A        TXA
9BBA    0A        ASL
9BBB    0A        ASL
9BBC    A8        TAY
9BBD    C908      CMP    #$08
9BBF    D017      BNE    $9BD8
9BC1    B5CB      LDA    $CB,X
9BC3    38        SEC
9BC4    E901      SBC    #$01
9BC6    C90E      CMP    #$0E
9BC8    9002      BCC    $9BCC
9BCA    A90E      LDA    #$0E
9BCC    0A        ASL
9BCD    0A        ASL
9BCE    B4CF      LDY    $CF,X
9BD0    F002      BEQ    $9BD4
9BD2    A981      LDA    #$81
9BD4    A008      LDY    #$08
9BD6    D006      BNE    $9BDE
9BD8    B5BC      LDA    $BC,X
9BDA    15D2      ORA    $D2,X
9BDC    15CF      ORA    $CF,X
9BDE    990040    STA    $4000,Y
9BE1    68        PLA
9BE2    38        SEC
9BE3    E941      SBC    #$41
9BE5    A8        TAY
9BE6    B5BF      LDA    $BF,X
9BE8    AA        TAX
9BE9    BDA780    LDA    $80A7,X
9BEC    18        CLC
9BED    79A080    ADC    $80A0,Y
9BF0    65C2      ADC    $C2
9BF2    AA        TAX
9BF3    A5CA      LDA    $CA
9BF5    0A        ASL
9BF6    0A        ASL
9BF7    A8        TAY
9BF8    C908      CMP    #$08
9BFA    D005      BNE    $9C01
9BFC    8A        TXA
9BFD    18        CLC
9BFE    6918      ADC    #$18
9C00    AA        TAX
9C01    BD0680    LDA    $8006,X
9C04    990240    STA    $4002,Y
9C07    BD0780    LDA    $8007,X
9C0A    990340    STA    $4003,Y
9C0D    60        RTS
;音名处理子程序
9C0E    B191      LDA    ($91),Y
9C10    F013      BEQ    $9C25
9C12    C93A      CMP    #$3A
9C14    B00F      BCS    $9C25
9C16    38        SEC
9C17    E930      SBC    #$30
9C19    900A      BCC    $9C25
9C1B    AA        TAX
9C1C    BD9680    LDA    $8096,X
9C1F    A6CA      LDX    $CA
9C21    95C3      STA    $C3,X
9C23    E691      INC    $91
```

```
9C25    20CDA1    JSR    $A1CD
9C28    A6CA      LDX    $CA
9C2A    B5C3      LDA    $C3,X
9C2C    8528      STA    $28
9C2E    B5B9      LDA    $B9,X
9C30    852C      STA    $2C
9C32    205C8B    JSR    $8B5C
9C35    A528      LDA    $28
9C37    A6CA      LDX    $CA
9C39    95CB      STA    $CB,X
9C3B    A591      LDA    $91
9C3D    95C6      STA    $C6,X
9C3F    60        RTS
;把 PLAY 语句中的内容装入 $700 以后
9C40    AD0004    LDA    $0400
9C43    F012      BEQ    $9C57
9C45    CE0004    DEC    $0400
9C48    209D8C    JSR    $8C9D
9C4B    C93A      CMP    #$3A
9C4D    F008      BEQ    $9C57
9C4F    A000      LDY    #$00
9C51    9191      STA    ($91),Y
9C53    E691      INC    $91
9C55    D0E9      BNE    $9C40
9C57    A000      LDY    #$00
9C59    A900      LDA    #$00
9C5B    9191      STA    ($91),Y
9C5D    60        RTS
9C5E    4C6D83    JMP    $836D
9C61    4C7083    JMP    $8370
9C64    20A18C    JSR    $8CA1
9C67    4C6D9C    JMP    $9C6D
9C6A    2094A5    JSR    $A594
9C6D    20829C    JSR    $9C82
9C70    20EFA3    JSR    $A3EF
9C73    AD0104    LDA    $0401
9C76    D0E9      BNE    $9C61
9C78    AD0004    LDA    $0400
9C7B    60        RTS
9C7C    20829C    JSR    $9C82
9C7F    4CF2A3    JMP    $A3F2
9C82    A900      LDA    #$00
9C84    8527      STA    $27
9C86    8530      STA    $30
9C88    A980      LDA    #$80
9C8A    8521      STA    $21
9C8C    20949C    JSR    $9C94
9C8F    A527      LDA    $27
9C91    F0CB      BEQ    $9C5E
9C93    60        RTS
9C94    20BD9C    JSR    $9CBD
9C97    C9EF      CMP    #$EF
9C99    F001      BEQ    $9C9C
9C9B    60        RTS
        XOR 处理入口
9C9C    20EFA3    JSR    $A3EF
9C9F    20A18C    JSR    $8CA1
9CA2    20BD9C    JSR    $9CBD
9CA5    20EFA3    JSR    $A3EF
9CA8    20E09D    JSR    $9DE0
9CAB    A528      LDA    $28
9CAD    452C      EOR    $2C
9CAF    8528      STA    $28
9CB1    A529      LDA    $29
9CB3    452D      EOR    $2D
9CB5    8529      STA    $29
9CB7    20D7A1    JSR    $A1D7
9CBA    4C979C    JMP    $9C97
9CBD    20E69C    JSR    $9CE6
9CC0    C9F0      CMP    #$F0
9CC2    F001      BEQ    $9CC5
9CC4    60        RTS
        OR 处理入口
9CC5    20EFA3    JSR    $A3EF
9CC8    20A18C    JSR    $8CA1
9CCB    20E69C    JSR    $9CE6
9CCE    20EFA3    JSR    $A3EF
9CD1    20E09D    JSR    $9DE0
9CD4    A528      LDA    $28
9CD6    052C      ORA    $2C
9CD8    8528      STA    $28
9CDA    A529      LDA    $29
9CDC    052D      ORA    $2D
9CDE    8529      STA    $29
9CE0    20D7A1    JSR    $A1D7
9CE3    4CC09C    JMP    $9CC0
9CE6    200F9D    JSR    $9D0F
9CE9    C9F1      CMP    #$F1
```

```
9CEB    F001      BEQ    $9CEE
9CED    60        RTS
  AND 处理入口
9CEE    20EFA3    JSR    $A3EF
9CF1    20A18C    JSR    $8CA1
9CF4    200F9D    JSR    $9D0F
9CF7    20EFA3    JSR    $A3EF
9CFA    20E09D    JSR    $9DE0
9CFD    A528      LDA    $28
9CFF    252C      AND    $2C
9D01    8528      STA    $28
9D03    A529      LDA    $29
9D05    252D      AND    $2D
9D07    8529      STA    $29
9D09    20D7A1    JSR    $A1D7
9D0C    4CE99C    JMP    $9CE9
9D0F    204E9D    JSR    $9D4E
9D12    C9F3      CMP    #$F3
9D14    B001      BCS    $9D17
9D16    60        RTS
  <>~< 的处理入口
9D17    C9F9      CMP    #$F9
9D19    B0FB      BCS    $9D16
9D1B    38        SEC
9D1C    E9F3      SBC    #$F3
9D1E    0A        ASL
9D1F    AA        TAX
9D20    BD7DA2    LDA    $A27D,X
9D23    48        PHA
9D24    BD7EA2    LDA    $A27E,X
9D27    48        PHA
9D28    A527      LDA    $27
9D2A    48        PHA
9D2B    20A18C    JSR    $8CA1
9D2E    204E9D    JSR    $9D4E
9D31    68        PLA
9D32    C527      CMP    $27
9D34    D050      BNE    $9D86
9D36    20E09D    JSR    $9DE0
9D39    68        PLA
9D3A    851A      STA    $1A
9D3C    68        PLA
9D3D    8519      STA    $19
9D3F    204B9D    JSR    $9D4B
9D42    48        PHA
9D43    A902      LDA    #$02
9D45    8527      STA    $27
9D47    68        PLA
9D48    4C129D    JMP    $9D12
9D4B    6C1900    JMP    ($0019)
9D4E    20899D    JSR    $9D89
9D51    C9F9      CMP    #$F9
9D53    F005      BEQ    $9D5A
9D55    C9FA      CMP    #$FA
9D57    F00A      BEQ    $9D63
9D59    60        RTS
  + 运算的处理
9D5A    A9D4      LDA    #$D4
9D5C    48        PHA
9D5D    A9A1      LDA    #$A1
9D5F    48        PHA
9D60    4C699D    JMP    $9D69
  - 运算的处理
9D63    A9EE      LDA    #$EE
9D65    48        PHA
9D66    A9A1      LDA    #$A1
9D68    48        PHA
9D69    A527      LDA    $27
9D6B    48        PHA
9D6C    20A18C    JSR    $8CA1
9D6F    20899D    JSR    $9D89
9D72    68        PLA
9D73    C527      CMP    $27
9D75    D00F      BNE    $9D86
9D77    20E09D    JSR    $9DE0
9D7A    68        PLA
9D7B    851A      STA    $1A
9D7D    68        PLA
9D7E    8519      STA    $19
9D80    204B9D    JSR    $9D4B
9D83    4C519D    JMP    $9D51
9D86    4CE1A3    JMP    $A3E1
9D89    20A89D    JSR    $9DA8
9D8C    C9FB      CMP    #$FB
9D8E    F001      BEQ    $9D91
9D90    60        RTS
  MOD 运算的处理
9D91    A527      LDA    $27
```

```
9D93   48       PHA
9D94   20A18C   JSR   $8CA1
9D97   20A89D   JSR   $9DA8
9D9A   68       PLA
9D9B   C527     CMP   $27
9D9D   D0E7     BNE   $9D86
9D9F   20E09D   JSR   $9DE0
9DA2   201DA2   JSR   $A21D
9DA5   4C8C9D   JMP   $9D8C
9DA8   20AB9E   JSR   $9EAB
9DAB   C9FC     CMP   #$FC
9DAD   F005     BEQ   $9DB4
9DAF   C9FD     CMP   #$FD
9DB1   F00A     BEQ   $9DBD
9DB3   60       RTS
   ÷ 运算的处理
9DB4   A9FA     LDA   #$FA
9DB6   48       PHA
9DB7   A9A1     LDA   #$A1
9DB9   48       PHA
9DBA   4CC39D   JMP   $9DC3
   × 运算的处理
9DBD   A9F4     LDA   #$F4
9DBF   48       PHA
9DC0   A9A1     LDA   #$A1
9DC2   48       PHA
9DC3   A527     LDA   $27
9DC5   48       PHA
9DC6   20A18C   JSR   $8CA1
9DC9   20AB9E   JSR   $9EAB
9DCC   68       PLA
9DCD   C527     CMP   $27
9DCF   D0B5     BNE   $9D86
9DD1   20E09D   JSR   $9DE0
9DD4   68       PLA
9DD5   851A     STA   $1A
9DD7   68       PLA
9DD8   8519     STA   $19
9DDA   204B9D   JSR   $9D4B
9DDD   4CAB9D   JMP   $9DAB
9DE0   A630     LDX   $30
9DE2   E07D     CPX   #$7D
9DE4   B03D     BCS   $9E23
9DE6   CA       DEX
9DE7   CA       DEX
9DE8   CA       DEX
9DE9   8630     STX   $30
9DEB   BD0004   LDA   $0400,X
9DEE   852C     STA   $2C
9DF0   BD0104   LDA   $0401,X
9DF3   852D     STA   $2D
9DF5   BD0240   LDA   $0402,X
9DF8   852E     STA   $2E
9DFA   CA       DEX
9DFB   CA       DEX
9DFC   CA       DEX
9DFD   BD0004   LDA   $0400,X
9E00   8528     STA   $28
9E02   BD0104   LDA   $0401,X
9E05   8529     STA   $29
9E07   BD0204   LDA   $0402,X
9E0A   952A     STA   $2A
9E0C   60       RTS
9E0D   20A18C   JSR   $8CA1
9E10   20AB9E   JSR   $9EAB
9E13   A630     LDX   $30
9E15   CA       DEX
9E16   CA       DEX
9E17   CA       DEX
9E18   BD0004   LDA   $0400,X
9E1B   8528     STA   $28
9E1D   BD0104   LDA   $0401,X
9E20   8529     STA   $29
9E22   60       RTS
;出错处理,代码=4
9E23   4C7083   JMP   $8370
   负数的处理
9E26   200D9E   JSR   $9E0D
9E29   207C8A   JSR   $8A7C
9E2C   A528     LDA   $28
9E2E   9D0004   STA   $0400,X
9E31   A529     LDA   $29
9E33   9D0104   STA   $0401,X
9E36   4CD48C   JMP   $8CD4
   NOT 运算的处理
9E39   200D9E   JSR   $9E0D
9E3C   A9FF     LDA   #$FF
9E3E   38       SEC
```

```
9E3F    FD0004    SBC    $0400,X
9E42    9D0004    STA    $0400,X
9E45    A9FF      LDA    #$FF
9E47    38        SEC
9E48    FD0104    SBC    $0401,X
9E4B    9D0104    STA    $0401,X
9E4E    A902      LDA    #$02
9E50    8527      STA    $27
9E52    60        RTS
```

读入字符串的处理

```
9E53    20A18C    JSR    $8CA1
9E56    A000      LDY    #$00
9E58    A621      LDX    $21
9E5A    B109      LDA    ($09),Y
9E5C    F013      BEQ    $9E71
9E5E    C922      CMP    #$22
;是引号则结束处理
9E60    F00C      BEQ    $9E6E
9E62    9D0007    STA    $0700,X
9E65    E8        INX
9E66    C8        INY
9E67    C020      CPY    #$20
9E69    F0B8      BEQ    $9E23
9E6B    4C5A9E    JMP    $9E5A
;引号中的字符串读取完毕的处理
9E6E    20A18C    JSR    $8CA1
9E71    8619      STX    $19
9E73    8490      STY    $90
9E75    98        TYA
9E76    18        CLC
9E77    6509      ADC    $09
9E79    8509      STA    $09
9E7B    A900      LDA    #$00
9E7D    650A      ADC    $0A
9E7F    850A      STA    $0A
9E81    A630      LDX    $30
9E83    A590      LDA    $90
9E85    9D0004    STA    $0400,X
9E88    A521      LDA    $21
9E8A    9D0104    STA    $0401,X
9E8D    A907      LDA    #$07
9E8F    9D0204    STA    $0402,X
9E92    E8        INX
9E93    E8        INX
9E94    E8        INX
9E95    8630      STX    $30
9E97    A519      LDA    $19
9E99    8521      STA    $21
9E9B    A903      LDA    #$03
9E9D    8527      STA    $27
9E9F    4CD48C    JMP    $8CD4
9EA2    4C6D83    JMP    $836D
9EA5    4C269E    JMP    $9E26
```

执行程序的处理

```
9EA8    20A18C    JSR    $8CA1
9EAB    207883    JSR    $8378
9EAE    F0F2      BEQ    $9EA2
9EB0    C9FA      CMP    #$FA
;减运算转$9EA5
9EB2    F0F1      BEQ    $9EA5
9EB4    C9F2      CMP    #$F2
;NOT 运算转$9E39
9EB6    F081      BEQ    $9E39
9EB8    C912      CMP    #$12
9EBA    F051      BEQ    $9F0D
9EBC    C911      CMP    #$11
;是数据转$9F0D
9EBE    F04D      BEQ    $9F0D
9EC0    C9F9      CMP    #$F9
;加运算转$9EA8
9EC2    F0E4      BEQ    $9EA8
9EC4    C928      CMP    #$28
;是"("转$9F01
9EC6    F039      BEQ    $9F01
9EC8    C922      CMP    #$22
;是引号转$9E53
9ECA    F087      BEQ    $9E53
9ECC    C941      CMP    #$41
9ECE    902B      BCC    $9EFB
9ED0    C95B      CMP    #$5B
;是大写字母转$9F2B
9ED2    9057      BCC    $9F2B
9ED4    C9A1      CMP    #$A1
;是 MOVE 转$9EFE
9ED6    F026      BEQ    $9EFE
9ED8    C9CA      CMP    #$CA
9EDA    901F      BCC    $9EFB
9EDC    C9EF      CMP    #$EF
```

```
9EDE    B01B      BCS    $9EFB
9EE0    38        SEC
9EE1    E9CA      SBC    #$CA
9EE3    0A        ASL
9EE4    AA        TAX
9EE5    BD28C3    LDA    $C328,X
9EE8    8519      STA    $19
9EEA    BD29C3    LDA    $C329,X
9EED    851A      STA    $1A
9EEF    20A18C    JSR    $8CA1
9EF2    20D48C    JSR    $8CD4
9EF5    204B9D    JSR    $9D4B
9EF8    4CD48C    JMP    $8CD4
;出错处理,代码=1
9EFB    4C6D83    JMP    $836D
;转 MOVE(n)的处理
9EFE    4C9DA2    JMP    $A29D
9F01    20A18C    JSR    $8CA1
9F04    20949C    JSR    $9C94
9F07    2093A2    JSR    $A293
9F0A    4CD48C    JMP    $8CD4
    程序中数据的处理
9F0D    A630      LDX    $30
9F0F    A001      LDY    #$01
9F11    B109      LDA    ($09),Y
9F13    9D0004    STA    $0400,X
9F16    C8        INY
9F17    E8        INX
9F18    B109      LDA    ($09),Y
9F1A    9D0004    STA    $0400,X
9F1D    E8        INX
9F1E    E8        INX
9F1F    8630      STX    $30
9F21    A902      LDA    #$02
9F23    8527      STA    $27
9F25    202799    JSR    $9927
9F28    4CD48C    JMP    $8CD4
    程序中变量的处理
9F2B    207E9F    JSR    $9F7E
9F2E    A630      LDX    $30
9F30    A590      LDA    $90
9F32    290F      AND    #$0F
9F34    C903      CMP    #$03
9F36    F021      BEQ    $9F59
;数值变量的处理
9F38    A000      LDY    #$00
9F3A    B18C      LDA    ($8C),Y
9F3C    9D0004    STA    $0400,X
9F3F    C8        INY
9F40    B18C      LDA    ($8C),Y
9F42    9D0104    STA    $0401,X
9F45    C8        INY
9F46    B18C      LDA    ($8C),Y
9F48    9D0204    STA    $0402,X
9F4B    E8        INX
9F4C    E8        INX
9F4D    E8        INX
9F4E    8630      STX    $30
9F50    A590      LDA    $90
9F52    290F      AND    #$0F
9F54    8527      STA    $27
9F56    4CD48C    JMP    $8CD4
    串变量的取值处理
9F59    A000      LDY    #$00
9F5B    B18C      LDA    ($8C),Y
9F5D    9D0004    STA    $0400,X
9F60    A901      LDA    #$01
9F62    18        CLC
9F63    658C      ADC    $8C
9F65    9D0104    STA    $0401,X
9F68    A900      LDA    #$00
9F6A    658D      ADC    $8D
9F6C    9D0204    STA    $0402,X
9F6F    4C4B9F    JMP    $9F4B
9F72    4C6D83    JMP    $836D
    变量的处理
;变量名存入$8A、$8B单元
变量类型存$90单元
变量值首址存入$$8C、$8D单元
9F75    20C3A1    JSR    $A1C3
9F78    B0F8      BCS    $9F72
9F7A    A200      LDX    #$00
9F7C    8630      STX    $30
9F7E    A200      LDX    #$00
9F80    868B      STX    $8B
9F82    858A      STA    $8A
9F84    20AD8C    JSR    $8CAD
9F87    20BBA1    JSR    $A1BB
```

```
9F8A    B00A      BCS    $9F96
9F8C    858B      STA    $8B
9F8E    20AD8C    JSR    $8CAD
9F91    20BBA1    JSR    $A1BB
9F94    90F8      BCC    $9F8E
9F96    A902      LDA    #$02
9F98    8590      STA    $90
9F9A    20D48C    JSR    $8CD4
9F9D    C924      CMP    #$24
9F9F    D007      BNE    $9FA8
;串变量的处理
9FA1    A903      LDA    #$03
9FA3    8590      STA    $90
9FA5    20A18C    JSR    $8CA1
9FA8    20D48C    JSR    $8CD4
9FAB    C928      CMP    #$28
9FAD    F070      BEQ    $A01F
;简单变量的检索处理
9FAF    200EA1    JSR    $A10E
9FB2    B045      BCS    $9FF9
9FB4    A000      LDY    #$00
9FB6    A590      LDA    $90
;建立简单变量区
第一个单元为变量类型标志
第二、三个单元为变量名
第四、五个单元为变量值
第六个单元为结束标志
最后置入三个零
9FB8    918E      STA    ($8E),Y
9FBA    C8        INY
9FBB    A58A      LDA    $8A
9FBD    918E      STA    ($8E),Y
9FBF    C8        INY
9FC0    A58B      LDA    $8B
9FC2    918E      STA    ($8E),Y
9FC4    C8        INY
9FC5    A900      LDA    #$00
9FC7    918E      STA    ($8E),Y
9FC9    C8        INY
9FCA    918E      STA    ($8E),Y
9FCC    C8        INY
9FCD    918E      STA    ($8E),Y
9FCF    A590      LDA    $90
9FD1    C902      CMP    #$02
9FD3    F00C      BEQ    $9FE1
;建立串变量区
9FD5    A900      LDA    #$00
9FD7    A221      LDX    #$21
9FD9    C8        INY
9FDA    918E      STA    ($8E),Y
9FDC    CA        DEX
9FDD    D0FA      BNE    $9FD9
9FDF    A921      LDA    #$21
9FE1    18        CLC
9FE2    6903      ADC    #$03
9FE4    658E      ADC    $8E
9FE6    8528      STA    $28
9FE8    A900      LDA    #$00
9FEA    658F      ADC    $8F
9FEC    8529      STA    $29
9FEE    20448A    JSR    $8A44
9FF1    A528      LDA    $28
9FF3    851F      STA    $1F
9FF5    A529      LDA    $29
9FF7    8520      STA    $20
;检索到变量的处理
9FF9    18        CLC
;令$$8C、$8D指向变量值首址
9FFA    A903      LDA    #$03
9FFC    658E      ADC    $8E
9FFE    858C      STA    $8C
A000    A900      LDA    #$00
A002    658F      ADC    $8F
A004    858D      STA    $8D
A006    60        RTS
A007    A58E      LDA    $8E
A009    858C      STA    $8C
A00B    A58F      LDA    $8F
A00D    858D      STA    $8D
A00F    60        RTS
A010    20949C    JSR    $9C94
A013    A902      LDA    #$02
A015    C527      CMP    $27
A017    F0F6      BEQ    $A00F
;出错处理，类型=TM
A019    4CE1A3    JMP    $A3E1
A01C    4CBCA0    JMP    $A0BC
        是下标变量的处理
```

```
A01F    A58A     LDA    $8A
A021    48       PHA
A022    A58B     LDA    $8B
A024    48       PHA
A025    A590     LDA    $90
A027    0980     ORA    #$80
A029    48       PHA
A02A    20A18C   JSR    $8CA1
A02D    2010A0   JSR    $A010
A030    A630     LDX    $30
A032    CA       DEX
A033    CA       DEX
A034    CA       DEX
A035    BD0004   LDA    $0400,X
A038    48       PHA
A039    20F1A0   JSR    $A0F1
A03C    90DE     BCC    $A01C
A03E    2094A5   JSR    $A594
A041    2010A0   JSR    $A010
A044    A630     LDX    $30
A046    CA       DEX
A047    CA       DEX
A048    CA       DEX
A049    BD0004   LDA    $0400,X
A04C    48       PHA
A04D    20F1A0   JSR    $A0F1
A050    B0C7     BCS    $A019
A052    20A18C   JSR    $8CA1
A055    68       PLA
A056    851B     STA    $1B
A058    68       PLA
A059    8519     STA    $19
A05B    68       PLA
A05C    8590     STA    $90
A05E    68       PLA
A05F    858B     STA    $8B
A061    68       PLA
A062    858A     STA    $8A
A064    200EA1   JSR    $A10E
A067    9050     BCC    $A0B9
A069    A003     LDY    #$03
A06B    B18E     LDA    ($8E),Y
A06D    F04A     BEQ    $A0B9
A06F    C8       INY
A070    B18E     LDA    ($8E),Y
A072    C519     CMP    $19
A074    9043     BCC    $A0B9
A076    AA       TAX
A077    C8       INY
A078    B18E     LDA    ($8E),Y
A07A    C51B     CMP    $1B
A07C    903B     BCC    $A0B9
A07E    20CDA1   JSR    $A1CD
A081    E8       INX
A082    D002     BNE    $A086
A084    E629     INC    $29
A086    8628     STX    $28
A088    A51B     LDA    $1B
A08A    852C     STA    $2C
A08C    205C8B   JSR    $8B5C
A08F    A519     LDA    $19
A091    852C     STA    $2C
A093    20E88A   JSR    $8AE8
A096    2003A1   JSR    $A103
A099    852C     STA    $2C
A09B    205C8B   JSR    $8B5C
A09E    A906     LDA    #$06
A0A0    852C     STA    $2C
A0A2    20E88A   JSR    $8AE8
A0A5    A58E     LDA    $8E
A0A7    852C     STA    $2C
A0A9    A58F     LDA    $8F
A0AB    852D     STA    $2D
A0AD    20E88A   JSR    $8AE8
A0B0    A528     LDA    $28
A0B2    858C     STA    $8C
A0B4    A529     LDA    $29
A0B6    858D     STA    $8D
A0B8    60       RTS
;出错处理,类型=IL
A0B9    4C7083   JMP    $8370
A0BC    20A18C   JSR    $8CA1
A0BF    68       PLA
A0C0    8519     STA    $19
A0C2    68       PLA
A0C3    8590     STA    $90
A0C5    68       PLA
A0C6    858B     STA    $8B
```

```
A0C8    68        PLA
A0C9    858A      STA    $8A
A0CB    200EA1    JSR    $A10E
A0CE    90E9      BCC    $A0B9
A0D0    A003      LDY    #$03
A0D2    B18E      LDA    ($8E),Y
A0D4    D0E3      BNE    $A0B9
A0D6    C8        INY
A0D7    B18E      LDA    ($8E),Y
A0D9    C519      CMP    $19
A0DB    90DC      BCC    $A0B9
A0DD    A519      LDA    $19
A0DF    8528      STA    $28
A0E1    2003A1    JSR    $A103
A0E4    852C      STA    $2C
A0E6    20CDA1    JSR    $A1CD
A0E9    205C8B    JSR    $8B5C
A0EC    A905      LDA    #$05
A0EE    4CA0A0    JMP    $A0A0
A0F1    BD0104    LDA    $0401,X
A0F4    D0C3      BNE    $A0B9
A0F6    8630      STX    $30
A0F8    20D48C    JSR    $8CD4
A0FB    C929      CMP    #$29
A0FD    D002      BNE    $A101
A0FF    18        CLC
A100    60        RTS
A101    38        SEC
A102    60        RTS
A103    A590      LDA    $90
A105    290F      AND    #$0F
A107    C902      CMP    #$02
A109    F002      BEQ    $A10D
A10B    A921      LDA    #$21
A10D    60        RTS
```

从变量区检索变量

```
A10E    A51D      LDA    $1D
A110    858E      STA    $8E
A112    A51E      LDA    $1E
A114    858F      STA    $8F
A116    A000      LDY    #$00
A118    B18E      LDA    ($8E),Y
A11A    F036      BEQ    $A152
A11C    C590      CMP    $90
;变量类型不同转$A130
A11E    D010      BNE    $A130
;检索到同类型变量的处理
A120    C8        INY
;比较变量名
A121    B18E      LDA    ($8E),Y
A123    C58A      CMP    $8A
A125    D009      BNE    $A130
;当变量名的第一个字母相同时再比较第二个字母
A127    C8        INY
A128    B18E      LDA    ($8E),Y
A12A    C58B      CMP    $8B
A12C    D002      BNE    $A130
A12E    38        SEC
A12F    60        RTS
```

未查到变量的处理

```
A130    A000      LDY    #$00
A132    B18E      LDA    ($8E),Y
A134    301E      BMI    $A154
A136    2007A1    JSR    $A107
A139    18        CLC
A13A    6903      ADC    #$03
A13C    658E      ADC    $8E
A13E    858E      STA    $8E
A140    A900      LDA    #$00
A142    658F      ADC    $8F
A144    858F      STA    $8F
A146    A51F      LDA    $1F
A148    C58E      CMP    $8E
A14A    D0CA      BNE    $A116
A14C    A520      LDA    $20
A14E    C58F      CMP    $8F
A150    D0C4      BNE    $A116
A152    18        CLC
A153    60        RTS
```

下标变量的检索处理

```
A154    290F      AND    #$0F
A156    8573      STA    $73
A158    A003      LDY    #$03
A15A    B18E      LDA    ($8E),Y
;取数组维数标志存入$74
A15C    8574      STA    $74
A15E    C8        INY
A15F    B18E      LDA    ($8E),Y
```

```
;取第一下标
A161    854A     STA   $4A
A163    C8       INY
A164    B18E     LDA   ($8E),Y
;取第二下标
A166    854B     STA   $4B
;计算下标变量数量存入$28、$29单元
A168    20CDA1   JSR   $A1CD
A16B    852C     STA   $2C
A16D    A54A     LDA   $4A
A16F    8528     STA   $28
A171    A574     LDA   $74
A173    F004     BEQ   $A179
A175    A54B     LDA   $4B
A177    852C     STA   $2C
A179    E628     INC   $28
A17B    D002     BNE   $A17F
A17D    E629     INC   $29
A17F    E62C     INC   $2C
A181    D002     BNE   $A185
A183    E62D     INC   $2D
A185    205C8B   JSR   $8B5C
A188    A573     LDA   $73
A18A    2007A1   JSR   $A107
A18D    852C     STA   $2C
A18F    A900     LDA   #$00
A191    852D     STA   $2D
A193    205C8B   JSR   $8B5C
A196    A905     LDA   #$05
A198    852C     STA   $2C
A19A    A574     LDA   $74
A19C    F002     BEQ   $A1A0
A19E    E62C     INC   $2C
A1A0    20E88A   JSR   $8AE8
A1A3    A58E     LDA   $8E
A1A5    852C     STA   $2C
A1A7    A58F     LDA   $8F
A1A9    852D     STA   $2D
A1AB    20E88A   JSR   $8AE8
A1AE    A528     LDA   $28
A1B0    858E     STA   $8E
A1B2    A529     LDA   $29
A1B4    858F     STA   $8F
A1B6    4C46A1   JMP   $A146
A1B9    38       SEC
A1BA    60       RTS
A1BB    C930     CMP   #$30
A1BD    90FA     BCC   $A1B9
A1BF    C93A     CMP   #$3A
A1C1    9008     BCC   $A1CB
A1C3    C941     CMP   #$41
A1C5    90F2     BCC   $A1B9
A1C7    C95B     CMP   #$5B
A1C9    B0EE     BCS   $A1B9
A1CB    18       CLC
A1CC    60       RTS
A1CD    A900     LDA   #$00
A1CF    852D     STA   $2D
A1D1    8529     STA   $29
A1D3    60       RTS
        除法运算处理
A1D4    20EC8A   JSR   $8AEC
A1D7    A630     LDX   $30
A1D9    CA       DEX
A1DA    CA       DEX
A1DB    CA       DEX
A1DC    A528     LDA   $28
A1DE    9D0004   STA   $0400,X
A1E1    A529     LDA   $29
A1E3    9D0104   STA   $0401,X
A1E6    A52A     LDA   $2A
A1E8    9D0204   STA   $0402,X
A1EB    4CD48C   JMP   $8CD4
A1EE    20738A   JSR   $8A73
A1F1    4CD7A1   JMP   $A1D7
;乘运算处理，乘积存入$28、$29
A1F4    20518B   JSR   $8B51
A1F7    4CD7A1   JMP   $A1D7
A1FA    A52C     LDA   $2C
A1FC    D004     BNE   $A202
A1FE    A52D     LDA   $2D
A200    F016     BEQ   $A218
A202    EA       NOP
A203    EA       NOP
A204    EA       NOP
A205    EA       NOP
A206    EA       NOP
;除运算处理，商存入$28、$29
```

```
A207    20B28B    JSR    $8BB2
A20A    4CD7A1    JMP    $A1D7
A20D    A901      LDA    #$01
A20F    8528      STA    $28
A211    A900      LDA    #$00
A213    8529      STA    $29
A215    4CD7A1    JMP    $A1D7
A218    A90A      LDA    #$0A
;出错处理，类型=DZ
A21A    4C8FA9    JMP    $A98F
A21D    A529      LDA    $29
A21F    48        PHA
;除运算子程序
A220    20FAA1    JSR    $A1FA
A223    A52A      LDA    $2A
A225    8528      STA    $28
A227    A52B      LDA    $2B
A229    8529      STA    $29
A22B    68        PLA
A22C    100A      BPL    $A238
A22E    A529      LDA    $29
A230    3003      BMI    $A235
A232    207C8A    JSR    $8A7C
A235    4CD7A1    JMP    $A1D7
A238    A529      LDA    $29
A23A    10F9      BPL    $A235
A23C    30F4      BMI    $A232
;比较两个操作数的大小
A23E    20038C    JSR    $8C03
A241    D009      BNE    $A24C
A243    A900      LDA    #$00
A245    8528      STA    $28
A247    8529      STA    $29
A249    4CD7A1    JMP    $A1D7
A24C    A9FF      LDA    #$FF
A24E    4C45A2    JMP    $A245
;比较两个操作数的大小
A251    20038C    JSR    $8C03
A254    F0F6      BEQ    $A24C
A256    90F4      BCC    $A24C
A258    4C43A2    JMP    $A243
;比较两个操作数的大小
A25B    20038C    JSR    $8C03
A25E    B0EC      BCS    $A24C
A260    4C43A2    JMP    $A243
;比较两个操作数的大小
A263    20038C    JSR    $8C03
A266    F0E4      BEQ    $A24C
A268    4C43A2    JMP    $A243
;比较两个操作数的大小
A26B    20038C    JSR    $8C03
A26E    F0D3      BEQ    $A243
A270    B0DA      BCS    $A24C
A272    4C43A2    JMP    $A243
;比较两个操作数的大小
A275    20038C    JSR    $8C03
A278    90D2      BCC    $A24C
A27A    4C43A2    JMP    $A243
A27D    3E A2  5B A2  51 A2  63 A2  6B A2  75 A2
          ≠      ≥      ≤      =      >      <
A289    20D48C    JSR    $8CD4
A28C    C928      CMP    #$28
A28E    D04E      BNE    $A2DE
A290    4CA18C    JMP    $8CA1
A293    20D48C    JSR    $8CD4
A296    C929      CMP    #$29
A298    D044      BNE    $A2DE
A29A    4CA18C    JMP    $8CA1
        MOVE(N)指令的解释程序入口
A29D    20A18C    JSR    $8CA1
A2A0    20E6A3    JSR    $A3E6
A2A3    A428      LDY    $28
A2A5    B94506    LDA    $0645,Y
A2A8    A000      LDY    #$00
A2AA    2910      AND    #$10
A2AC    F002      BEQ    $A2B0
A2AE    A0FF      LDY    #$FF
A2B0    8428      STY    $28
A2B2    8429      STY    $29
A2B4    2063A3    JSR    $A363
A2B7    4CD48C    JMP    $8CD4
A2BA    A900      LDA    #$00
A2BC    8528      STA    $28
A2BE    8529      STA    $29
A2C0    852A      STA    $2A
A2C2    60        RTS
        读括号中的数据子程序
A2C3    2089A2    JSR    $A289
```

```
A2C6   20949C   JSR   $9C94
A2C9   A630     LDX   $30
A2CB   CA       DEX
A2CC   BD0004   LDA   $0400,X
A2CF   852A     STA   $2A
A2D1   CA       DEX
A2D2   BD0004   LDA   $0400,X
A2D5   8529     STA   $29
A2D7   CA       DEX
A2D8   BD0004   LDA   $0400,X
A2DB   8528     STA   $28
A2DD   60       RTS
A2DE   4C6D83   JMP   $836D
A2E1   A630     LDX   $30
A2E3   A528     LDA   $28
A2E5   9D0004   STA   $0400,X
A2E8   E8       INX
A2E9   A529     LDA   $29
A2EB   9D0004   STA   $0400,X
A2EE   E8       INX
A2EF   E8       INX
A2F0   8630     STX   $30
A2F2   60       RTS
      CSRLIN 保留字入口
A2F3   A549     LDA   $49
A2F5   8528     STA   $28
A2F7   A900     LDA   #$00
A2F9   8529     STA   $29
A2FB   E630     INC   $30
A2FD   E630     INC   $30
A2FF   E630     INC   $30
A301   A902     LDA   #$02
A303   8527     STA   $27
A305   A630     LDX   $30
A307   CA       DEX
A308   CA       DEX
A309   CA       DEX
A30A   A528     LDA   $28
A30C   9D0004   STA   $0400,X
A30F   A529     LDA   $29
A311   9D0104   STA   $0401,X
A314   A52A     LDA   $2A
A316   9D0204   STA   $0402,X
A319   60       RTS
      FRE 保留字入口
A31A   2020A3   JSR   $A320
A31D   4CFBA2   JMP   $A2FB
A320   38       SEC
A321   A503     LDA   $03
A323   E51F     SBC   $1F
A325   8528     STA   $28
A327   A504     LDA   $04
A329   E520     SBC   $20
A32B   8529     STA   $29
A32D   60       RTS
      POS 保留字入口
A32E   20ECA3   JSR   $A3EC
A331   A548     LDA   $48
A333   8528     STA   $28
A335   A900     LDA   #$00
A337   8529     STA   $29
A339   4C63A3   JMP   $A363
A33C   20E6A3   JSR   $A3E6
A33F   A528     LDA   $28
A341   0A       ASL
A342   0A       ASL
A343   0A       ASL
A344   AA       TAX
A345   60       RTS
      XPOS 保留字入口
;读卡通编号
A346   203CA3   JSR   $A33C
A349   BDC606   LDA   $06C6,X
A34C   4C33A3   JMP   $A333
      YPOS 保留字入口
;读卡通编号
A34F   203CA3   JSR   $A33C
A352   BDC706   LDA   $06C7,X
A355   4C33A3   JMP   $A333
      PEEK 保留字入口
;读出括号中的数据存入$28、$29中
A358   20ECA3   JSR   $A3EC
A35B   A000     LDY   #$00
A35D   B128     LDA   ($28),Y
A35F   8528     STA   $28
A361   8429     STY   $29
A363   2093A2   JSR   $A293
A366   4C01A3   JMP   $A301
```

```
    ABS 保留字入口
;读出括号中的数据存入 $28、$29 中
A369    20ECA3    JSR    $A3EC
A36C    A529      LDA    $29
A36E    3003      BMI    $A373
A370    4C63A3    JMP    $A363
A373    207C8A    JSR    $8A7C
A376    4C63A3    JMP    $A363
    SGN 保留字入口
;读出括号中的数据存入 $28、$29 中
A379    20ECA3    JSR    $A3EC
A37C    A529      LDA    $29
A37E    300D      BMI    $A38D
A380    F014      BEQ    $A396
A382    A900      LDA    #$00
A384    8529      STA    $29
A386    A901      LDA    #$01
A388    8528      STA    $28
A38A    4C63A3    JMP    $A363
A38D    A9FF      LDA    #$FF
A38F    8528      STA    $28
A391    8529      STA    $29
A393    4C63A3    JMP    $A363
A396    A528      LDA    $28
A398    D0E8      BNE    $A382
A39A    A900      LDA    #$00
A39C    4C8FA3    JMP    $A38F
;读出括号中的数据存入 $28、$29 中
A39F    20ECA3    JSR    $A3EC
A3A2    A528      LDA    $28
A3A4    48        PHA
A3A5    A529      LDA    $29
A3A7    48        PHA
A3A8    A556      LDA    $56
A3AA    8528      STA    $28
A3AC    A557      LDA    $57
A3AE    8529      STA    $29
A3B0    A983      LDA    #$83
A3B2    852C      STA    $2C
A3B4    A903      LDA    #$03
A3B6    852D      STA    $2D
;乘法运算，积存于 $28、$29
A3B8    20598B    JSR    $8B59
A3BB    A97F      LDA    #$7F
A3BD    2529      AND    $29
A3BF    8529      STA.   $29
A3C1    8557      STA    $57
A3C3    A528      LDA    $28
A3C5    8556      STA    $56
A3C7    68        PLA
A3C8    852D      STA    $2D
A3CA    68        PLA
A3CB    852C      STA    $2C
;乘法运算，积存于 $28、$29
A3CD    20598B    JSR    $8B59
A3D0    0629      ASL    $29
A3D2    2636      ROL    $36
A3D4    2637      ROL    $37
A3D6    A536      LDA    $36
A3D8    8528      STA    $28
A3DA    A537      LDA    $37
A3DC    8529      STA    $29
A3DE    4C63A3    JMP    $A363
A3E1    A90B      LDA    #$0B
;出错处理，类型=TM
A3E3    4C8FA9    JMP    $A98F
A3E6    20ECA3    JSR    $A3EC
A3E9    4C48A5    JMP    $A548
;读出括号中的数据存入 $28、$29 中
A3EC    20C3A2    JSR    $A2C3
;数据类型判断
A3EF    A902      LDA    #$02
A3F1    AEA903    LDX    $03A9
A3F4    C527      CMP    $27
A3F6    D0E9      BNE    $A3E1
A3F8    60        RTS
A3F9    20C3A2    JSR    $A2C3
A3FC    4CF2A3    JMP    $A3F2
    LEN 保留字入口
A3FF    20F9A3    JSR    $A3F9
A402    A900      LDA    #$00
A404    20BEA2    JSR    $A2BE
A407    4C63A3    JMP    $A363
    ASC 保留字入口
A40A    20F9A3    JSR    $A3F9
A40D    A528      LDA    $28
A40F    F0F1      BEQ    $A402
A411    A000      LDY    #$00
```

```
A413    B129      LDA   ($29),Y
A415    8528      STA   $28
A417    4C02A4    JMP   $A402
;出错处理，类型=IL
A41A    4C7083    JMP   $8370
    SCR$保留字入口
A41D    20A1AC    JSR   $ACA1
A420    20E6A3    JSR   $A3E6
A423    A528      LDA   $28
A425    C91C      CMP   #$1C
A427    B0F1      BCS   $A41A
A429    8548      STA   $48
A42B    209EA5    JSR   $A59E
A42E    2048A5    JSR   $A548
A431    A528      LDA   $28
A433    C918      CMP   #$18
A435    B0E3      BCS   $A41A
A437    8549      STA   $49
A439    20D48C    JSR   $8CD4
A43C    C929      CMP   #$29
A43E    F00F      BEQ   $A44F
;SCR$(X,Y,n)语句的处理
A440    209EA5    JSR   $A59E
A443    A528      LDA   $28
A445    F008      BEQ   $A44F
A447    2037BA    JSR   $BA37
A44A    A59C      LDA   $9C
A44C    4C59A4    JMP   $A459
;SCR$(X,Y)语句的处理
A44F    20A6AB    JSR   $ABA6
A452    A905      LDA   #$05
A454    200888    JSR   $8808
A457    A573      LDA   $73
A459    8528      STA   $28
A45B    20AAAC    JSR   $ACAA
A45E    4C64A4    JMP   $A464
    CHR$保留字入口
A461    20E6A3    JSR   $A3E6
A464    A528      LDA   $28
A466    2073A4    JSR   $A473
A469    A903      LDA   #$03
A46B    8527      STA   $27
A46D    2093A2    JSR   $A293
A470    4C05A3    JMP   $A305
A473    A621      LDX   $21
A475    9D0007    STA   $0700,X
A478    E621      INC   $21
A47A    A901      LDA   #$01
A47C    8528      STA   $28
A47E    8629      STX   $29
A480    A907      LDA   #$07
A482    852A      STA   $2A
A484    60        RTS
    STR$保留字入口
;读出括号中的数据存入$28、$29
A485    20ECA3    JSR   $A3EC
;把数据转换为ASCII码存入$3D~$41单元
A488    20968D    JSR   $8D96
A48B    20EB8C    JSR   $8CEB
A48E    A900      LDA   #$00
A490    8590      STA   $90
A492    A421      LDY   $21
A494    A646      LDX   $46
A496    B53D      LDA   $3D,X
A498    F00A      BEQ   $A4A4
A49A    990007    STA   $0700,Y
A49D    E8        INX
A49E    C8        INY
A49F    E690      INC   $90
A4A1    4C96A4    JMP   $A496
A4A4    A521      LDA   $21
A4A6    8421      STY   $21
A4A8    8529      STA   $29
A4AA    A907      LDA   #$07
A4AC    852A      STA   $2A
A4AE    A590      LDA   $90
A4B0    8528      STA   $28
A4B2    4C69A4    JMP   $A469
    HEX$保留字入口
;读出括号中的数据存入$28、$29
A4B5    20ECA3    JSR   $A3EC
;把数据转换为ASCII码存入$3D~$41单元
A4B8    20D88D    JSR   $8DD8
A4BB    4C8BA4    JMP   $A48B
    VAL保留字入口
A4BE    20F9A3    JSR   $A3F9
A4C1    A528      LDA   $28
A4C3    F027      BEQ   $A4EC
```

```
A4C5    20F88F    JSR    $8FF8
A4C8    A528      LDA    $28
A4CA    8574      STA    $74
A4CC    A200      LDX    #$00
A4CE    A000      LDY    #$00
A4D0    B129      LDA    ($29),Y
A4D2    9D0005    STA    $0500,X
A4D5    E8        INX
A4D6    C8        INY
A4D7    C674      DEC    $74
A4D9    D0F5      BNE    $A4D0
A4DB    A900      LDA    #$00
A4DD    9D0005    STA    $0500,X
A4E0    20328A    JSR    $8A32
A4E3    20FEA4    JSR    $A4FE
A4E6    200190    JSR    $9001
A4E9    4C63A3    JMP    $A363
A4EC    20BAA2    JSR    $A2BA
A4EF    4C63A3    JMP    $A363
;负数的转换处理
A4F2    20A18C    JSR    $8CA1
A4F5    20FEA4    JSR    $A4FE
A4F8    4C7C8A    JMP    $8A7C
A4FB    20A18C    JSR    $8CA1
A4FE    A000      LDY    #$00
A500    B109      LDA    ($09),Y
A502    C92D      CMP    #$2D
A504    F0EC      BEQ    $A4F2
A506    C92B      CMP    #$2B
A508    F0F1      BEQ    $A4FB
A50A    C926      CMP    #$26
A50C    F003      BEQ    $A511
A50E    4CF98C    JMP    $8CF9
;十六进制数据的转换处理
A511    20A18C    JSR    $8CA1
A514    B109      LDA    ($09),Y
A516    C948      CMP    #$48
A518    D003      BNE    $A51D
A51A    20A18C    JSR    $8CA1
A51D    4CEB8D    JMP    $8DEB
```

INKEY$ 保留字入口

```
A520    207883    JSR    $8378
A523    F00D      BEQ    $A532
A525    C928      CMP    #$28
A527    D009      BNE    $A532
;INKEY$(0)的处理
A529    20C3A2    JSR    $A2C3
A52C    2063B2    JSR    $B263
A52F    4C66A4    JMP    $A466
;INKEY$的处理
A532    2088AF    JSR    $AF88
A535    F00B      BEQ    $A542
A537    2073A4    JSR    $A473
A53A    20FBA2    JSR    $A2FB
A53D    A903      LDA    #$03
A53F    8527      STA    $27
A541    60        RTS
A542    20BAA2    JSR    $A2BA
A545    4C3AA5    JMP    $A53A
A548    A529      LDA    $29
A54A    D006      BNE    $A552
A54C    60        RTS
A54D    C920      CMP    #$20
A54F    B001      BCS    $A552
A551    60        RTS
;出错处理，类型=IL
A552    4C7083    JMP    $8370
A555    20F9A3    JSR    $A3F9
;字符函数中括号中内容的处理子程序
A558    A528      LDA    $28
A55A    48        PHA
A55B    A529      LDA    $29
A55D    48        PHA
A55E    A52A      LDA    $2A
A560    48        PHA
A561    209EA5    JSR    $A59E
A564    2048A5    JSR    $A548
A567    A528      LDA    $28
A569    204DA5    JSR    $A54D
A56C    68        PLA
A56D    851A      STA    $1A
A56F    68        PLA
A570    8519      STA    $19
A572    68        PLA
A573    60        RTS
```

LEFT$ 保留字入口

```
;取括号中串变量值的地址
A574    2055A5    JSR    $A555
```

```
A577  854A    STA  $4A
A579  A528    LDA  $28
A57B  C54A    CMP  $4A
A57D  B00B    BCS  $A58A
A57F  A519    LDA  $19
A581  8529    STA  $29
A583  A51A    LDA  $1A
A585  852A    STA  $2A
A587  4C69A4  JMP  $A469
A58A  A54A    LDA  $4A
A58C  8528    STA  $28
A58E  4C7FA5  JMP  $A57F
;出错处理，类型=SN
A591  4C6D83  JMP  $836D
A594  20D48C  JSR  $8CD4
A597  C92C    CMP  #$2C
A599  D0F6    BNE  $A591
A59B  4CA18C  JMP  $8CA1
A59E  2094A5  JSR  $A594
A5A1  20949C  JSR  $9C94
A5A4  A630    LDX  $30
A5A6  CA      DEX
A5A7  BD0004  LDA  $0400,X
A5AA  852A    STA  $2A
A5AC  CA      DEX
A5AD  BD0004  LDA  $0400,X
A5B0  8529    STA  $29
A5B2  CA      DEX
A5B3  BD0004  LDA  $0400,X
A5B6  8528    STA  $28
A5B8  8630    STX  $30
A5BA  4CEFA3  JMP  $A3EF
    RIGHT$保留只入口
;取括号中串变量值的地址
A5BD  2055A5  JSR  $A555
A5C0  854A    STA  $4A
A5C2  A528    LDA  $28
A5C4  C54A    CMP  $4A
A5C6  B0C2    BCS  $A58A
A5C8  38      SEC
A5C9  A54A    LDA  $4A
A5CB  E528    SBC  $28
A5CD  18      CLC
A5CE  6519    ADC  $19
A5D0  8519    STA  $19
A5D2  A900    LDA  #$00
A5D4  651A    ADC  $1A
A5D6  851A    STA  $1A
A5D8  A528    LDA  $28
A5DA  8590    STA  $90
A5DC  A621    LDX  $21
A5DE  A000    LDY  #$00
A5E0  B119    LDA  ($19),Y
A5E2  9D0007  STA  $0700,X
A5E5  E8      INX
A5E6  C8      INY
A5E7  C690    DEC  $90
A5E9  D0F5    BNE  $A5E0
A5EB  A521    LDA  $21
A5ED  8529    STA  $29
A5EF  8621    STX  $21
A5F1  A907    LDA  #$07
A5F3  852A    STA  $2A
A5F5  4C69A4  JMP  $A469
;出错处理，类型=IL
A5F8  4C7083  JMP  $8370
    MID$保留字入口
;检索串变量
A5FB  20F9A3  JSR  $A3F9
A5FE  A528    LDA  $28
A600  48      PHA
A601  A529    LDA  $29
A603  48      PHA
A604  A52A    LDA  $2A
A606  48      PHA
A607  209EA5  JSR  $A59E
A60A  2048A5  JSR  $A548
A60D  A528    LDA  $28
A60F  F0E7    BEQ  $A5F8
A611  204DA5  JSR  $A54D
A614  38      SEC
A615  E901    SBC  #$01
A617  48      PHA
A618  209EA5  JSR  $A59E
A61B  2048A5  JSR  $A548
A61E  A528    LDA  $28
A620  204DA5  JSR  $A54D
A623  851C    STA  $1C
```

```
A625    68        PLA
A626    851B      STA    $1B
A628    68        PLA
A629    851A      STA    $1A
A62B    68        PLA
A62C    8519      STA    $19
A62E    68        PLA
A62F    C51B      CMP    $1B
A631    903B      BCC    $A66E
A633    38        SEC
A634    E51B      SBC    $1B
A636    C51C      CMP    $1C
A638    902F      BCC    $A669
A63A    A51B      LDA    $1B
A63C    18        CLC
A63D    6519      ADC    $19
A63F    8519      STA    $19
A641    A900      LDA    #$00
A643    651A      ADC    $1A
A645    851A      STA    $1A
;从字符串的第n位取m个字符
A647    A51C      LDA    $1C
A649    8528      STA    $28
A64B    8590      STA    $90
A64D    A621      LDX    $21
A64F    A000      LDY    #$00
A651    B119      LDA    ($19),Y
A653    9D0007    STA    $0700,X
A656    E8        INX
A657    C8        INY
A658    C690      DEC    $90
A65A    D0F5      BNE    $A651
A65C    A521      LDA    $21
A65E    8529      STA    $29
A660    8621      STX    $21
A662    A907      LDA    #$07
A664    852A      STA    $2A
A666    4C69A4    JMP    $A469
A669    851C      STA    $1C
A66B    4C3AA6    JMP    $A63A
A66E    20BAA2    JSR    $A2BA
A671    4C69A4    JMP    $A469
A674    4C7083    JMP    $8370
;读操纵器子程序
A677    20E6A3    JSR    $A3E6
A67A    A528      LDA    $28
A67C    F007      BEQ    $A685
A67E    C901      CMP    #$01
A680    D0F2      BNE    $A674
;读0#操纵器
A682    A917      LDA    #$17
;读1#操纵器
A684    AEA916    LDX    $16A9
A687    8519      STA    $19
A689    A940      LDA    #$40
A68B    851A      STA    $1A
A68D    A905      LDA    #$05
A68F    8D1640    STA    $4016
A692    EA        NOP
A693    EA        NOP
A694    A904      LDA    #$04
A696    8D1640    STA    $4016
A699    EA        NOP
A69A    EA        NOP
A69B    A208      LDX    #$08
A69D    A000      LDY    #$00
A69F    B119      LDA    ($19),Y
A6A1    4A        LSR
A6A2    2690      ROL    $90
A6A4    EA        NOP
A6A5    EA        NOP
A6A6    CA        DEX
A6A7    D0F6      BNE    $A69F
A6A9    60        RTS
```

STICK 保留字入口

```
A6AA    2077A6    JSR    $A677
A6AD    207AA6    JSR    $A67A
A6B0    A590      LDA    $90
A6B2    290F      AND    #$0F
A6B4    8528      STA    $28
A6B6    4C63A3    JMP    $A363
```

STRIG 保留字入口

```
A6B9    2077A6    JSR    $A677
A6BC    A590      LDA    $90
A6BE    4A        LSR
A6BF    4A        LSR
A6C0    4A        LSR
A6C1    4A        LSR
```

```
A6C2    4CB4A6    JMP    $A6B4
A6C5    0A        ASL
A6C6    0A        ASL
A6C7    0A        ASL
A6C8    0A        ASL
A6C9    60        RTS
   CGEN 保留字入口
A6CA    206D9C    JSR    $9C6D
A6CD    2903      AND    #$03
A6CF    0A        ASL
A6D0    0A        ASL
A6D1    0A        ASL
A6D2    8574      STA    $74
A6D4    A532      LDA    $32
A6D6    29E7      AND    #$E7
A6D8    0574      ORA    $74
A6DA    8532      STA    $32
A6DC    4CF1B3    JMP    $B3F1
A6DF    A200      LDX    #$00
A6E1    A9EF      LDA    #$EF
A6E3    9D0002    STA    $0200,X
A6E6    E8        INX
A6E7    D0FA      BNE    $A6E3
A6E9    60        RTS
   PALET 保留字入口
A6EA    A200      LDX    #$00
A6EC    8665      STX    $65
A6EE    C942      CMP    #$42
;是 PALETB 指令转$A6FA
A6F0    F008      BEQ    $A6FA
A6F2    C953      CMP    #$53
A6F4    D04D      BNE    $A743
;PALETS 指令的处理
A6F6    A910      LDA    #$10
A6F8    8565      STA    $65
A6FA    20649C    JSR    $9C64
A6FD    C904      CMP    #$04
A6FF    B03F      BCS    $A740
A701    48        PHA
A702    A904      LDA    #$04
A704    8502      STA    $02
A706    A900      LDA    #$00
A708    857C      STA    $7C
A70A    206A9C    JSR    $9C6A
A70D    A47C      LDY    $7C
A70F    993D00    STA    $003D,Y
A712    E67C      INC    $7C
A714    C602      DEC    $02
A716    D0F2      BNE    $A70A
A718    68        PLA
;配色代码存$700
C1→$701 C2→$702
C3→$703 C4→$704
A719    8D0007    STA    $0700
A71C    A53D      LDA    $3D
A71E    8D0107    STA    $0701
A721    A53E      LDA    $3E
A723    8D0207    STA    $0702
A726    A53F      LDA    $3F
A728    8D0307    STA    $0703
A72B    A540      LDA    $40
A72D    8D0407    STA    $0704
A730    A006      LDY    #$06
A732    A665      LDX    $65
A734    F001      BEQ    $A737
A736    C8        INY
A737    98        TYA
A738    4C0888    JMP    $8808
A73B    C908      CMP    #$08
A73D    B001      BCS    $A740
A73F    60        RTS
;出错处理，类型=IL
A740    4C7083    JMP    $8370
;出错处理，类型=SN
A743    4C6D83    JMP    $836D
   CGSET 保留字入口
A746    C92C      CMP    #$2C
A748    F025      BEQ    $A76F
;读背景配色代码
A74A    206D9C    JSR    $9C6D
A74D    F004      BEQ    $A753
A74F    C901      CMP    #$01
A751    D0ED      BNE    $A740
A753    0A        ASL
A754    AA        TAX
A755    BDA5C3    LDA    $C3A5,X
A758    8552      STA    $52
A75A    BDA6C3    LDA    $C3A6,X
```

```
A75D    8553      STA   $53
A75F    A900      LDA   #$00
A761    8565      STA   $65
A763    208AA7    JSR   $A78A
A766    207883    JSR   $8378
A769    F01E      BEQ   $A789
A76B    C92C      CMP   #$2C
A76D    D0D4      BNE   $A743
;读卡通配色代码
A76F    20649C    JSR   $9C64
A772    C903      CMP   #$03
A774    B0CA      BCS   $A740
A776    0A        ASL
A777    AA        TAX
A778    BDA9C3    LDA   $C3A9,X
A77B    8552      STA   $52
A77D    BDAAC3    LDA   $C3AA,X
A780    8553      STA   $53
A782    A901      LDA   #$01
A784    8565      STA   $65
A786    208AA7    JSR   $A78A
A789    60        RTS
    置PPU配色处理，把配色代码送入PPU
A78A    A900      LDA   #$00
A78C    8D0007    STA   $0700
A78F    8575      STA   $75
A791    A904      LDA   #$04
A793    A201      LDX   #$01
A795    8574      STA   $74
A797    A475      LDY   $75
A799    B152      LDA   ($52),Y
A79B    9D0007    STA   $0700,X
A79E    E8        INX
A79F    C8        INY
A7A0    C674      DEC   $74
A7A2    D0F5      BNE   $A799
A7A4    8475      STY   $75
A7A6    A906      LDA   #$06
A7A8    8573      STA   $73
A7AA    A565      LDA   $65
A7AC    F002      BEQ   $A7B0
A7AE    E673      INC   $73
A7B0    A573      LDA   $73
A7B2    200888    JSR   $8808
A7B5    EE0007    INC   $0700
A7B8    AD0007    LDA   $0700
A7BB    C904      CMP   #$04
A7BD    D0D2      BNE   $A791
A7BF    60        RTS
    SPRITE ON 的处理
A7C0    A533      LDA   $33
A7C2    0910      ORA   #$10
A7C4    8533      STA   $33
A7C6    8D0120    STA   $2001
A7C9    4CA18C    JMP   $8CA1
    SPRITE OFF 的处理
A7CC    A533      LDA   $33
A7CE    29EF      AND   #$EF
A7D0    4CC4A7    JMP   $A7C4
    SPRITE N 的处理
A7D3    68        PLA
A7D4    20C5A6    JSR   $A6C5
A7D7    AA        TAX
A7D8    A004      LDY   #$04
A7DA    A9EF      LDA   #$EF
A7DC    9D0002    STA   $0200,X
A7DF    9D0302    STA   $0203,X
A7E2    9D0102    STA   $0201,X
A7E5    E8        INX
A7E6    E8        INX
A7E7    E8        INX
A7E8    E8        INX
A7E9    88        DEY
A7EA    D0EE      BNE   $A7DA
A7EC    60        RTS
;出错处理，类型=IL;
A7ED    4C7083    JMP   $8370
;出错处理，类型=SN
A7F0    4C6D83    JMP   $836D
    SPRITE 保留字的入口
A7F3    C99A      CMP   #$9A
A7F5    F0C9      BEQ   $A7C0
A7F7    C99B      CMP   #$9B
A7F9    F0D1      BEQ   $A7CC
    SPRITE N,X,Y 的处理
A7FB    206D9C    JSR   $9C6D
A7FE    203BA7    JSR   $A73B
A801    48        PHA
```

```
A802    207883    JSR    $8378
A805    F0CC      BEQ    $A7D3
;取 X 座标
A807    206A9C    JSR    $9C6A
A80A    48        PHA
;取 Y 座标
A80B    206A9C    JSR    $9C6A
A80E    851B      STA    $1B
A810    68        PLA
A811    8519      STA    $19
A813    68        PLA
A814    8575      STA    $75
A816    AA        TAX
A817    BD3D06    LDA    $063D,X
A81A    8590      STA    $90
A81C    A575      LDA    $75
A81E    0A        ASL
A81F    0A        ASL
A820    AA        TAX
A821    0A        ASL
A822    0A        ASL
A823    48        PHA
A824    A8        TAY
A825    A904      LDA    #$04
A827    8574      STA    $74
A829    BD8006    LDA    $0680,X
A82C    990102    STA    $0201,Y
A82F    E8        INX
A830    C8        INY
A831    C8        INY
A832    C8        INY
A833    C8        INY
A834    C674      DEC    $74
;送卡通编号
A836    D0F1      BNE    $A829
A838    68        PLA
A839    A8        TAY
A83A    A51B      LDA    $1B
A83C    990002    STA    $0200,Y
A83F    A519      LDA    $19
A841    990302    STA    $0203,Y
A844    A590      LDA    $90
;送第一块卡通块的座标
A846    D001      BNE    $A849
A848    60        RTS
A849    A900      LDA    #$00
A84B    851C      STA    $1C
A84D    A908      LDA    #$08
A84F    851A      STA    $1A
;送第二块卡通块的座标
A851    2065A8    JSR    $A865
A854    A900      LDA    #$00
A856    851A      STA    $1A
A858    A908      LDA    #$08
A85A    851C      STA    $1C
;送第三块卡通块的座标
A85C    2065A8    JSR    $A865
A85F    A908      LDA    #$08
A861    851A      STA    $1A
;送第四块卡通块的座标
A863    851C      STA    $1C
;送卡通块的座标子程序
A865    C8        INY
A866    C8        INY
A867    C8        INY
A868    C8        INY
A869    A519      LDA    $19
A86B    18        CLC
A86C    651A      ADC    $1A
A86E    990302    STA    $0203,Y
A871    A51B      LDA    $1B
A873    18        CLC
A874    651C      ADC    $1C
A876    990002    STA    $0200,Y
A879    60        RTS
A87A    C9A1      CMP    #$A1
A87C    D003      BNE    $A881
A87E    4CE4BF    JMP    $BFE4
A881    4C6D83    JMP    $836D
A884    4C7083    JMP    $8370
        DEF 保留字的入口
A887    C98A      CMP    #$8A
A889    D0EF      BNE    $A87A
;DEF SPRITE 指令的处理
A88B    20649C    JSR    $9C64
A88E    203BA7    JSR    $A73B
A891    48        PHA
A892    2094A5    JSR    $A594
```

```
A895  2089A2   JSR   $A289
A898  A904     LDA   #$04
A89A  8575     STA   $75
A89C  206D9C   JSR   $9C6D
A89F  48       PHA
;读出 DEF SPRITE 语句中的参数入栈
A8A0  206A9C   JSR   $9C6A
A8A3  48       PHA
A8A4  C675     DEC   $75
A8A6  D0F8     BNE   $A8A0
A8A8  2093A2   JSR   $A293
A8AB  20A285   JSR   $85A2
;Y 轴翻转
A8AE  68       PLA
A8AF  6A       ROR
A8B0  6A       ROR
A8B1  2980     AND   #$80
A8B3  8519     STA   $19
;X 轴翻转
A8B5  68       PLA
A8B6  6A       ROR
A8B7  6A       ROR
A8B8  6A       ROR
A8B9  2940     AND   #$40
A8BB  0519     ORA   $19
A8BD  8519     STA   $19
;显示优先度
A8BF  68       PLA
A8C0  6A       ROR
A8C1  6A       ROR
A8C2  6A       ROR
A8C3  6A       ROR
A8C4  2920     AND   #$20
A8C6  0519     ORA   $19
A8C8  8519     STA   $19
;配色
A8CA  68       PLA
A8CB  85AD     STA   $AD
;编号
A8CD  68       PLA
A8CE  2903     AND   #$03
A8D0  0519     ORA   $19
A8D2  8519     STA   $19
A8D4  68       PLA
A8D5  8575     STA   $75
A8D7  20C5A6   JSR   $A6C5
A8DA  A004     LDY   #$04
A8DC  AA       TAX
A8DD  A900     LDA   #$00
A8DF  9D0002   STA   $0200,X
A8E2  9D0302   STA   $0203,X
A8E5  A519     LDA   $19
A8E7  9D0202   STA   $0202,X
A8EA  A920     LDA   #$20
A8EC  9D0102   STA   $0201,X
A8EF  E8       INX
A8F0  E8       INX
A8F1  E8       INX
A8F2  E8       INX
A8F3  88       DEY
A8F4  D0E7     BNE   $A8DD
;取 DEF SPRITE 语句等号后的内容
A8F6  207C9C   JSR   $9C7C
A8F9  AD0104   LDA   $0401
A8FC  851B     STA   $1B
A8FE  AD0204   LDA   $0402
A901  851C     STA   $1C
A903  A000     LDY   #$00
A905  A575     LDA   $75
A907  0A       ASL
A908  0A       ASL
A909  AA       TAX
A90A  AD0004   LDA   $0400
A90D  F010     BEQ   $A91F
A90F  B11B     LDA   ($1B),Y
A911  9D8006   STA   $0680,X
A914  E8       INX
A915  C8       INY
A916  C004     CPY   #$04
A918  F010     BEQ   $A92A
A91A  CE0004   DEC   $0400
A91D  D0F0     BNE   $A90F
A91F  A920     LDA   #$20
A921  9D8006   STA   $0680,X
A924  E8       INX
A925  C8       INY
A926  C004     CPY   #$04
A928  D0F7     BNE   $A921
```

```
A92A   A675     LDX   $75
A92C   A5AD     LDA   $AD
A92E   9D3D06   STA   $063D,X
A931   60       RTS
    VIEW 保留字的入口
;清屏
A932   2075AB   JSR   $AB75
A935   A920     LDA   #$20
A937   8574     STA   $74
A939   A90C     LDA   #$0C
A93B   200888   JSR   $8808
;把背景一页的内容送入零页
A93E   A200     LDX   #$00
A940   A574     LDA   $74
A942   18       CLC
A943   6904     ADC   #$04
A945   20F881   JSR   $81F8
A948   AD0720   LDA   $2007
A94B   AD0720   LDA   $2007
A94E   48       PHA
A94F   A574     LDA   $74
A951   20F881   JSR   $81F8
A954   68       PLA
A955   8D0720   STA   $2007
A958   E8       INX
A959   D0E5     BNE   $A940
A95B   E674     INC   $74
A95D   A574     LDA   $74
A95F   C924     CMP   #$24
A961   D0DD     BNE   $A940
A963   20E6AB   JSR   $ABE6
A966   20F1B3   JSR   $B3F1
A969   A90B     LDA   #$0B
A96B   4C0888   JMP   $8808
    六页单元初始化处理
A96E   A23D     LDX   #$3D
A970   A900     LDA   #$00
A972   9D0006   STA   $0600,X
A975   E8       INX
A976   D0FA     BNE   $A972
A978   A200     LDX   #$00
A97A   A9EF     LDA   #$EF
A97C   9D8006   STA   $0680,X
A97F   E8       INX
A980   E020     CPX   #$20
A982   D0F8     BNE   $A97C
A984   A2C0     LDX   #$C0
A986   A901     LDA   #$01
A988   9D0006   STA   $0600,X
A98B   E8       INX
A98C   D0FA     BNE   $A988
A98E   60       RTS
    出错处理子程序
A98F   48       PHA
A990   2003AC   JSR   $AC03
A993   68       PLA
A994   A200     LDX   #$00
A996   9A       TXS
A997   0A       ASL
A998   A8       TAY
A999   A93F     LDA   #$3F
A99B   20F087   JSR   $87F0
A99E   B95EAA   LDA   $AA5E,Y
;送显错误类型的第一字符
A9A1   20F087   JSR   $87F0
A9A4   B95FAA   LDA   $AA5F,Y
;送显错误类型的第二字符
A9A7   20F087   JSR   $87F0
A9AA   A93F     LDA   #$3F
A9AC   8552     STA   $52
A9AE   A9AA     LDA   #$AA
A9B0   8553     STA   $53
;送显 ERROR
A9B2   20B787   JSR   $87B7
A9B5   203B87   JSR   $873B
A9B8   A9FF     LDA   #$FF
A9BA   C50C     CMP   $0C
A9BC   D004     BNE   $A9C2
A9BE   C50B     CMP   $0B
A9C0   F00E     BEQ   $A9D0
A9C2   A934     LDA   #$34
A9C4   8552     STA   $52
A9C6   A9AA     LDA   #$AA
A9C8   8553     STA   $53
;送显 IN
A9CA   20B787   JSR   $87B7
;送显出错行号
A9CD   20D9A9   JSR   $A9D9
```

```
A9D0    20FB87    JSR    $87FB
;发一声蜂鸣声
A9D3    200FB3    JSR    $B30F
A9D6    4C2182    JMP    $8221
    把当前行号转换为十进制数存入
    $3D～$41
A9D9    A50B      LDA    $0B
A9DB    8528      STA    $28
A9DD    A50C      LDA    $0C
A9DF    8529      STA    $29
A9E1    203C8D    JSR    $8D3C
A9E4    20EB8C    JSR    $8CEB
A9E7    4C5F86    JMP    $865F
    送显按 STOP 键后的 IN BREAK......
A9EA    2003AC    JSR    $AC03
A9ED    A900      LDA    #$00
A9EF    8562      STA    $62
A9F1    A939      LDA    #$39
A9F3    8552      STA    $52
A9F5    A9AA      LDA    #$AA
A9F7    8553      STA    $53
A9F9    20B787    JSR    $87B7
A9FC    4CB8A9    JMP    $A9B8
    送显 OK
A9FF    A94F      LDA    #$4F
AA01    200DAC    JSR    $AC0D
AA04    A94B      LDA    #$4B
AA06    4C0DAC    JMP    $AC0D
    送显 F BASIC BS. 2A......
AA09    A958      LDA    #$58
AA0B    8552      STA    $52
AA0D    A9C3      LDA    #$C3
AA0F    8553      STA    $53
AA11    20F887    JSR    $87F8
AA14    2020A3    JSR    $A320
AA17    20E1A9    JSR    $A9E1
AA1A    20FB87    JSR    $87FB
AA1D    A928      LDA    #$28
AA1F    8552      STA    $52
AA21    A9AA      LDA    #$AA
AA23    8553      STA    $53
AA25    4CF887    JMP    $87F8
    复位画面数据区
AA28                            20 42 59 54 45 53 20 46
                                   B  Y  T  E  S     F
AA30    52 45 45 00 20 49 4E 20 00 42 52 45 41 4B 00 20
        R  E  E        I  N        B  R  E  A  K
AA40    45 52 52 4F 52 00 57 52 49 54 49 4E 47 20 00 4C
        E  R  R  O  R     W  R  I  T  I  N  G        L
AA50    4F 41 44 49 4E 47 20 00 53 4B 49 50 20 00 4E 46
        O  A  D  I  N  G        S  K  I  P        N  F
AA60    53 4E 52 47 4F 44 49 4C 4F 56 4F 4D 55 4C 53 4F
         SN    RG    OD    IL    IV    OM    UL    SO
AA70    44 44 44 5A 54 4D 53 54 46 54 43 43 55 46 4D 4F
         DD    DZ    TM    ST    FT    CC    UF    MO
AA80    54 50
         TP
    第一次进入 F BASIC 时对有关单元初始化
AA82    A910      LDA    #$10
AA84    8532      STA    $32
AA86    A90E      LDA    #$0E
AA88    8533      STA    $33
AA8A    20F1AB    JSR    $ABF1
AA8D    A900      LDA    #$00
AA8F    859D      STA    $9D
AA91    8569      STA    $69
AA93    8563      STA    $63
AA95    85AB      STA    $AB
AA97    85AC      STA    $AC
AA99    8562      STA    $62
AA9B    8572      STA    $72
AA9D    8550      STA    $50
AA9F    8518      STA    $18
```

```
AAA1   8D1040   STA   $4010
AAA4   8D0140   STA   $4001
AAA7   8D0540   STA   $4005
AAAA   8D1540   STA   $4015
AAAD   854C     STA   $4C
AAAF   8566     STA   $66
AAB1   8567     STA   $67
AAB3   A94F     LDA   #$4F
AAB5   8D1740   STA   $4017
AAB8   20F3AA   JSR   $AAF3
AABB   20F1B3   JSR   $B3F1
AABE   A920     LDA   #$20
AAC0   854F     STA   $4F
;清屏
AAC2   2075AB   JSR   $AB75
;送配色代码
AAC5   2009AB   JSR   $AB09
AAC8   A200     LDX   #$00
AACA   8A       TXA
AACB   A080     LDY   #$80
;清功能键定义区
AACD   9D8004   STA   $0480,X
AAD0   E8       INX
AAD1   88       DEY
AAD2   D0F9     BNE   $AACD
    定义功能键
AAD4   A200     LDX   #$00
AAD6   A567     LDA   $67
AAD8   C908     CMP   #$08
AADA   F02C     BEQ   $AB08
AADC   20C5A6   JSR   $A6C5
AADF   A8       TAY
AAE0   BD1DAB   LDA   $AB1D,X
AAE3   F008     BEQ   $AAED
AAE5   998004   STA   $0480,Y
AAE8   E8       INX
AAE9   C8       INY
AAEA   4CE0AA   JMP   $AAE0
AAED   E8       INX
AAEE   E667     INC   $67
AAF0   4CD6AA   JMP   $AAD6
    设定光标形状
AAF3   A9FD     LDA   #$FD
AAF5   85E0     STA   $E0
AAF7   A95F     LDA   #$5F
AAF9   85E1     STA   $E1
AAFB   4CA9AD   JMP   $ADA9
AAFE   AD0220   LDA   $2002
AB01   30FB     BMI   $AAFE
AB03   AD0220   LDA   $2002
AB06   10FB     BPL   $AB03
AB08   60       RTS
AB09   A9BF     LDA   #$BF
AB0B   8552     STA   $52
AB0D   A9C3     LDA   #$C3
AB0F   8553     STA   $53
AB11   A900     LDA   #$00
AB13   8565     STA   $65
AB15   208AA7   JSR   $A78A
AB18   E665     INC   $65
AB1A   4C8AA7   JMP   $A78A
    F BASIC 功能键数据区
AB1D                                            4C 4F 41
                                                L  O  A
AB20   44 0D 00 50 52 49 4E 54 20 00 47 4F 54 4F 20 00
       D           P  R  I  N  T        G  O  T  O
AB30   43 48 52 24 28 00 53 50 52 49 54 45 20 00 43 4F
       C  H  R  $  (     S  P  R  I  T  E        C  O
AB40   4E 54 0D 00 4C 49 53 54 0D 00 52 55 4E 0D 00
       N  T        L  I  S  T        R  U  N
    清屏处理
AB4F   A004     LDY   #$04
AB51   A2C0     LDX   #$C0
AB53   A920     LDA   #$20
AB55   8D0720   STA   $2007
AB58   CA       DEX
AB59   D0FA     BNE   $AB55
AB5B   88       DEY
```

```
AB5C    D0F7      BNE    $AB55
AB5E    A2C0      LDX    #$C0
AB60    A54F      LDA    $4F
AB62    18        CLC
AB63    6903      ADC    #$03
AB65    20F881    JSR    $81F8
AB68    A900      LDA    #$00
AB6A    8D0720    STA    $2007
AB6D    E8        INX
AB6E    D0FA      BNE    $AB6A
AB70    60        RTS
```

CLS 保留字入口

```
AB71    A924      LDA    #$24
AB73    854F      STA    $4F
AB75    A90C      LDA    #$0C
AB77    200888    JSR    $8808
AB7A    20EBB3    JSR    $B3EB
AB7D    20F1AB    JSR    $ABF1
AB80    A54F      LDA    $4F
AB82    A200      LDX    #$00
AB84    20F881    JSR    $81F8
```

;清屏

```
AB87    204FAB    JSR    $AB4F
AB8A    20E6AB    JSR    $ABE6
AB8D    20F1B3    JSR    $B3F1
AB90    A90B      LDA    #$0B
AB92    200888    JSR    $8808
AB95    208BAD    JSR    $AD8B
AB98    8569      STA    $69
AB9A    859D      STA    $9D
AB9C    AA        TAX
AB9D    9D0006    STA    $0600,X
ABA0    E8        INX
ABA1    E01A      CPX    #$1A
ABA3    D0F8      BNE    $AB9D
ABA5    60        RTS
```

计算光标对应的 PPU 地址存入 $4A、$4B

```
ABA6    A920      LDA    #$20
ABA8    8528      STA    $28
ABAA    A900      LDA    #$00
ABAC    8529      STA    $29
ABAE    852D      STA    $2D
ABB0    A569      LDA    $69
ABB2    4A        LSR
ABB3    4A        LSR
ABB4    4A        LSR
ABB5    18        CLC
ABB6    6549      ADC    $49
ABB8    6903      ADC    #$03
ABBA    C91E      CMP    #$1E
ABBC    9002      BCC    $ABC0
ABBE    E91E      SBC    #$1E
ABC0    852C      STA    $2C
ABC2    205C8B    JSR    $8B5C
ABC5    18        CLC
ABC6    A548      LDA    $48
ABC8    6902      ADC    #$02
ABCA    6528      ADC    $28
ABCC    854A      STA    $4A
ABCE    A900      LDA    #$00
ABD0    6529      ADC    $29
ABD2    18        CLC
ABD3    654F      ADC    $4F
ABD5    854B      STA    $4B
ABD7    60        RTS
ABD8    A54B      LDA    $4B
ABDA    8D0620    STA    $2006
ABDD    A54A      LDA    $4A
ABDF    8D0620    STA    $2006
ABE2    60        RTS
```

画面定位处理

```
ABE3    8D0720    STA    $2007
ABE6    A59D      LDA    $9D
ABE8    8D0520    STA    $2005
ABEB    A569      LDA    $69
ABED    8D0520    STA    $2005
ABF0    60        RTS
```

关显示

```
ABF1    20FEAA    JSR    $AAFE
ABF4    A900      LDA    #$00
ABF6    8D0120    STA    $2001
ABF9    60        RTS
```

开显示

```
ABFA    20FEAA    JSR    $AAFE
ABFD    A533      LDA    $33
ABFF    8D0120    STA    $2001
AC02    60        RTS
```

令光标下降一行

```
AC03    A548      LDA    $48
AC05    F0FB      BEQ    $AC02
AC07    A900      LDA    #$00
AC09    8548      STA    $48
AC0B    A90A      LDA    #$0A
```
编辑键判断处理
```
AC0D    C920      CMP    #$20
AC0F    B023      BCS    $AC34
AC11    C908      CMP    #$08
AC13    F010      BEQ    $AC25
AC15    C901      CMP    #$01
AC17    F00C      BEQ    $AC25
AC19    C912      CMP    #$12
AC1B    F008      BEQ    $AC25
AC1D    C909      CMP    #$09
AC1F    F004      BEQ    $AC25
AC21    A000      LDY    #$00
AC23    844C      STY    $4C
AC25    0A        ASL
AC26    AA        TAX
AC27    BD01AF    LDA    $AF01,X
AC2A    8528      STA    $28
AC2C    BD02AF    LDA    $AF02,X
AC2F    8529      STA    $29
AC31    6C2800    JMP    ($0028)
```
按 Ins 键的处理
```
AC34    A44C      LDY    $4C
AC36    F007      BEQ    $AC3F
AC38    48        PHA
AC39    A912      LDA    #$12
AC3B    200DAC    JSR    $AC0D
AC3E    68        PLA
AC3F    20F087    JSR    $87F0
AC42    4CFB87    JMP    $87FB
AC45    E649      INC    $49
AC47    A549      LDA    $49
AC49    C918      CMP    #$18
AC4B    F010      BEQ    $AC5D
AC4D    60        RTS
AC4E    208FAD    JSR    $AD8F
AC51    A649      LDX    $49
AC53    BD0106    LDA    $0601,X
AC56    F0ED      BEQ    $AC45
AC58    E649      INC    $49
AC5A    4C51AC    JMP    $AC51
```
屏幕上滚处理
```
AC5D    C67E      DEC    $7E
AC5F    208BAD    JSR    $AD8B
AC62    20A6AB    JSR    $ABA6
AC65    A904      LDA    #$04
AC67    200888    JSR    $8808
AC6A    A532      LDA    $32
AC6C    8D0020    STA    $2000
AC6F    A569      LDA    $69
AC71    18        CLC
AC72    6908      ADC    #$08
AC74    C9F0      CMP    #$F0
AC76    9002      BCC    $AC7A
AC78    A900      LDA    #$00
AC7A    8569      STA    $69
AC7C    20FEAA    JSR    $AAFE
AC7F    20E6AB    JSR    $ABE6
AC82    A018      LDY    #$18
AC84    A200      LDX    #$00
AC86    BD0206    LDA    $0602,X
AC89    9D0106    STA    $0601,X
AC8C    E8        INX
AC8D    88        DEY
AC8E    D0F6      BNE    $AC86
AC90    A900      LDA    #$00
AC92    8D1806    STA    $0618
AC95    8D1906    STA    $0619
AC98    8548      STA    $48
AC9A    A917      LDA    #$17
AC9C    8549      STA    $49
AC9E    4CF1B3    JMP    $B3F1
ACA1    A548      LDA    $48
ACA3    856B      STA    $6B
ACA5    A549      LDA    $49
ACA7    856C      STA    $6C
ACA9    60        RTS
ACAA    A56B      LDA    $6B
ACAC    8548      STA    $48
ACAE    A56C      LDA    $6C
ACB0    8549      STA    $49
ACB2    60        RTS
ACB3    20F1AB    JSR    $ABF1
ACB6    A910      LDA    #$10
```

```
ACB8   8532     STA   $32
ACBA   20F1B3   JSR   $B3F1
ACBD   A90E     LDA   #$0E
ACBF   8D0120   STA   $2001
ACC2   8533     STA   $33
ACC4   2009AB   JSR   $AB09
ACC7   205EAB   JSR   $AB5E
ACCA   854C     STA   $4C
ACCC   20E6AB   JSR   $ABE6
ACCF   4CFAAB   JMP   $ABFA
   按 ESC 键的处理
ACD2   20A1AC   JSR   $ACA1
ACD5   20A6AB   JSR   $ABA6
ACD8   A904     LDA   #$04
ACDA   200888   JSR   $8808
ACDD   A649     LDX   $49
ACDF   BD0106   LDA   $0601,X
ACE2   F00D     BEQ   $ACF1
ACE4   E649     INC   $49
ACE6   208FAD   JSR   $AD8F
ACE9   20A6AB   JSR   $ABA6
ACEC   A904     LDA   #$04
ACEE   4CDAAC   JMP   $ACDA
ACF1   4CAAAC   JMP   $ACAA
   按 Ctrl+Z 键的处理
ACF4   20A1AC   JSR   $ACA1
ACF7   20A6AB   JSR   $ABA6
ACFA   A904     LDA   #$04
ACFC   200888   JSR   $8808
ACFF   A649     LDX   $49
AD01   A900     LDA   #$00
AD03   9D0106   STA   $0601,X
AD06   8548     STA   $48
AD08   E649     INC   $49
AD0A   E8       INX
AD0B   E018     CPX   #$18
AD0D   F0E2     BEQ   $ACF1
AD0F   20A6AB   JSR   $ABA6
AD12   4CFAAC   JMP   $ACFA
AD15   A649     LDX   $49
AD17   CA       DEX
AD18   8A       TXA
AD19   AA       TAX
AD1A   E8       INX
AD1B   BD0106   LDA   $0601,X
AD1E   60       RTS
AD1F   4C07AC   JMP   $AC07
   按 Tab 键的处理
AD22   A548     LDA   $48
;令光标定位于 H07、H0E、H15、H1C 处
AD24   C915     CMP   #$15
AD26   B00B     BCS   $AD33
AD28   C90E     CMP   #$0E
AD2A   B00A     BCS   $AD36
AD2C   C907     CMP   #$07
AD2E   B009     BCS   $AD39
AD30   A907     LDA   #$07
AD32   ECA91C   CPX   $1CA9
AD35   ECA915   CPX   $15A9
AD38   ECA90E   CPX   $0EA9
AD3B   38       SEC
AD3C   E548     SBC   $48
AD3E   8562     STA   $62
AD40   AA       TAX
AD41   A920     LDA   #$20
AD43   9D0007   STA   $0700,X
AD46   CA       DEX
AD47   10FA     BPL   $AD43
AD49   4CFB87   JMP   $87FB
AD4C   A9FF     LDA   #$FF
AD4E   454C     EOR   $4C
AD50   854C     STA   $4C
AD52   60       RTS
   按 ↑ 键的处理
AD53   A549     LDA   $49
AD55   F002     BEQ   $AD59
AD57   C649     DEC   $49
AD59   60       RTS
   按 ↓ 键的处理
AD5A   A548     LDA   $48
AD5C   48       PHA
AD5D   2094AD   JSR   $AD94
AD60   68       PLA
AD61   8548     STA   $48
AD63   60       RTS
   按 ← 键的处理
AD64   A548     LDA   $48
AD66   F003     BEQ   $AD6B
```

```
AD68    C648      DEC   $48
AD6A    60        RTS
AD6B    A549      LDA   $49
AD6D    F006      BEQ   $AD75
AD6F    A91B      LDA   #$1B
AD71    8548      STA   $48
AD73    C649      DEC   $49
AD75    60        RTS
```

按 → 键的处理

```
AD76    A548      LDA   $48
AD78    C91B      CMP   #$1B
AD7A    F003      BEQ   $AD7F
AD7C    E648      INC   $48
AD7E    60        RTS
AD7F    208FAD    JSR   $AD8F
AD82    A549      LDA   $49
AD84    C917      CMP   #$17
AD86    F002      BEQ   $AD8A
AD88    E649      INC   $49
AD8A    60        RTS
```

按 Home 键的处理

```
AD8B    A900      LDA   #$00
AD8D    8549      STA   $49
AD8F    A900      LDA   #$00
AD91    8548      STA   $48
AD93    60        RTS
AD94    A649      LDX   $49
AD96    E8        INX
AD97    E018      CPX   #$18
AD99    D003      BNE   $AD9E
AD9B    4C5DAC    JMP   $AC5D
AD9E    8649      STX   $49
ADA0    60        RTS
```

按 CapsLock 键的处理

```
ADA1    A9FF      LDA   #$FF
ADA3    8572      STA   $72
ADA5    A5E1      LDA   $E1
ADA7    D006      BNE   $ADAF
ADA9    A900      LDA   #$00
ADAB    8572      STA   $72
ADAD    A5E0      LDA   $E0
ADAF    8561      STA   $61
ADB1    60        RTS
```

按 Del 键的处理

```
ADB2    A648      LDX   $48
ADB4    F005      BEQ   $ADBB
ADB6    C648      DEC   $48
ADB8    4CCBAD    JMP   $ADCB
ADBB    A649      LDX   $49
ADBD    F00C      BEQ   $ADCB
ADBF    CA        DEX
ADC0    BD0106    LDA   $0601,X
ADC3    F006      BEQ   $ADCB
ADC5    C649      DEC   $49
ADC7    A91B      LDA   #$1B
ADC9    8548      STA   $48
ADCB    20A1AC    JSR   $ACA1
ADCE    A548      LDA   $48
ADD0    C91B      CMP   #$1B
ADD2    F01B      BEQ   $ADEF
;计算光标位置
ADD4    20A6AB    JSR   $ABA6
ADD7    A54A      LDA   $4A
ADD9    8519      STA   $19
ADDB    A54B      LDA   $4B
ADDD    851A      STA   $1A
ADDF    E648      INC   $48
;计算光标位置
ADE1    20A6AB    JSR   $ABA6
ADE4    C648      DEC   $48
ADE6    A903      LDA   #$03
ADE8    200888    JSR   $8808
ADEB    A91B      LDA   #$1B
ADED    8548      STA   $48
ADEF    A649      LDX   $49
ADF1    BD0106    LDA   $0601,X
ADF4    D011      BNE   $AE07
;计算光标位置
ADF6    20A6AB    JSR   $ABA6
ADF9    20FEAA    JSR   $AAFE
ADFC    20D8AB    JSR   $ABD8
ADFF    A920      LDA   #$20
AE01    20E3AB    JSR   $ABE3
AE04    4CAAAC    JMP   $ACAA
AE07    E649      INC   $49
AE09    A9FF      LDA   #$FF
AE0B    8548      STA   $48
AE0D    2018AE    JSR   $AE18
```

```
AE10    E649      INC    $49
AE12    208FAD    JSR    $AD8F
AE15    4CCEAD    JMP    $ADCE
;令光标后的内容移到光标处
AE18    E648      INC    $48
;计算光标位置
AE1A    20A6AB    JSR    $ABA6
AE1D    20FEAA    JSR    $AAFE
AE20    20D8AB    JSR    $ABD8
AE23    AD0720    LDA    $2007
AE26    AD0720    LDA    $2007
AE29    48        PHA
AE2A    20E6AB    JSR    $ABE6
AE2D    C648      DEC    $48
AE2F    A548      LDA    $48
AE31    C9FF      CMP    #$FF
AE33    D006      BNE    $AE3B
AE35    A91B      LDA    #$1B
AE37    8548      STA    $48
AE39    C649      DEC    $49
AE3B    20A6AB    JSR    $ABA6
AE3E    20FEAA    JSR    $AAFE
AE41    20D8AB    JSR    $ABD8
AE44    68        PLA
AE45    4CE3AB    JMP    $ABE3
AE48    20A1AC    JSR    $ACA1
AE4B    A920      LDA    #$20
AE4D    85B0      STA    $B0
;计算光标位置
AE4F    20A6AB    JSR    $ABA6
AE52    A90D      LDA    #$0D
AE54    200888    JSR    $8808
    按 Del 键的处理
AE57    A649      LDX    $49
AE59    BD0106    LDA    $0601,X
AE5C    F00B      BEQ    $AE69
AE5E    E649      INC    $49
AE60    208FAD    JSR    $AD8F
AE63    4C4FAE    JMP    $AE4F
AE66    4CAAAC    JMP    $ACAA
AE69    A5B0      LDA    $B0
AE6B    C920      CMP    #$20
AE6D    F0F7      BEQ    $AE66
AE6F    A549      LDA    $49
AE71    C917      CMP    #$17
AE73    F061      BEQ    $AED6
AE75    208FAD    JSR    $AD8F
AE78    A449      LDY    $49
AE7A    C8        INY
AE7B    C017      CPY    #$17
AE7D    F026      BEQ    $AEA5
AE7F    A549      LDA    $49
AE81    8591      STA    $91
AE83    A916      LDA    #$16
AE85    8549      STA    $49
;计算光标位置
AE87    20A6AB    JSR    $ABA6
AE8A    A549      LDA    $49
AE8C    48        PHA
AE8D    A90E      LDA    #$0E
AE8F    200888    JSR    $8808
AE92    68        PLA
AE93    8549      STA    $49
AE95    AA        TAX
AE96    BD0106    LDA    $0601,X
AE99    E8        INX
AE9A    9D0106    STA    $0601,X
AE9D    C649      DEC    $49
AE9F    A549      LDA    $49
AEA1    C591      CMP    $91
AEA3    D0E2      BNE    $AE87
AEA5    A649      LDX    $49
AEA7    A9FF      LDA    #$FF
AEA9    9D0106    STA    $0601,X
AEAC    E649      INC    $49
AEAE    E8        INX
AEAF    A900      LDA    #$00
AEB1    9D0106    STA    $0601,X
AEB4    A217      LDX    #$17
AEB6    9D0106    STA    $0601,X
AEB9    A56B      LDA    $6B
AEBB    48        PHA
AEBC    A56C      LDA    $6C
AEBE    48        PHA
AEBF    A54C      LDA    $4C
AEC1    48        PHA
AEC2    A905      LDA    #$05
AEC4    200DAC    JSR    $AC0D
```

```
AEC7    68        PLA
AEC8    854C      STA    $4C
AECA    A90D      LDA    #$0D
AECC    200888    JSR    $8808
AECF    68        PLA
AED0    8549      STA    $49
AED2    68        PLA
AED3    8548      STA    $48
AED5    60        RTS
AED6    A56B      LDA    $6B
;令画面上滚一行的处理
AED8    48        PHA
AED9    A66C      LDX    $6C
AEDB    F001      BEQ    $AEDE
AEDD    CA        DEX
AEDE    8A        TXA
AEDF    48        PHA
AEE0    205DAC    JSR    $AC5D
AEE3    A917      LDA    #$17
AEE5    8549      STA    $49
AEE7    A900      LDA    #$00
AEE9    8548      STA    $48
AEEB    20A6AB    JSR    $ABA6
AEEE    A90D      LDA    #$0D
AEF0    200888    JSR    $8808
AEF3    A216      LDX    #$16
AEF5    A9FF      LDA    #$FF
AEF7    9D0106    STA    $0601,X
AEFA    68        PLA
AEFB    8549      STA    $49
AEFD    68        PLA
AEFE    8548      STA    $48
AF00    60        RTS
```

按键转移处理入口

```
AF01    4D AC 4C AD 4D AC 4E AC B3 AC D2 AC 4D AC 0F
AF10    B3 B2 AD 22 AD 94 AD 8B AD 75 AB 8F AD 4D AC 4D
AF20    AC 4D AC 4D AC 48 AE 4D AC 4D AC 4D AC A1 AD A9
AF30    AD 4D AC 4D AC F4 AC 4D AC 76 AD 64 AD 53 AD 5A
AF40    AD
```

功能键处理程序

```
AF41    A900      LDA    #$00
AF43    60        RTS
AF44    856D      STA    $6D
AF46    20DCB0    JSR    $B0DC
AF49    B005      BCS    $AF50
AF4B    A56D      LDA    $6D
AF4D    29FF      AND    #$FF
AF4F    60        RTS
AF50    A56D      LDA    $6D
AF52    A668      LDX    $68
AF54    D0EB      BNE    $AF41
AF56    856A      STA    $6A
AF58    A904      LDA    #$04
AF5A    8574      STA    $74
AF5C    209BAF    JSR    $AF9B
AF5F    C56A      CMP    $6A
AF61    D025      BNE    $AF88
AF63    C674      DEC    $74
AF65    D0F5      BNE    $AF5C
AF67    38        SEC
AF68    E9F0      SBC    #$F0
AF6A    20C5A6    JSR    $A6C5
AF6D    8567      STA    $67
AF6F    A9FF      LDA    #$FF
AF71    8566      STA    $66
AF73    A667      LDX    $67
AF75    BD8004    LDA    $0480,X
AF78    F006      BEQ    $AF80
AF7A    E8        INX
AF7B    8667      STX    $67
AF7D    29FF      AND    #$FF
AF7F    60        RTS
AF80    A900      LDA    #$00
AF82    8566      STA    $66
AF84    A2FF      LDX    #$FF
AF86    8668      STX    $68
```

读键控制程序

```
AF88    A566      LDA    $66
AF8A    D0E7      BNE    $AF73
AF8C    209BAF    JSR    $AF9B
```

```
AF8F  F007     BEQ  $AF98
AF91  C9F0     CMP  #$F0
AF93  B0AF     BCS  $AF44
AF95  29FF     AND  #$FF
AF97  60       RTS
AF98  8568     STA  $68
AF9A  60       RTS
AF9B  A905     LDA  #$05
AF9D  8D1640   STA  $4016
AFA0  EA       NOP
AFA1  EA       NOP
AFA2  EA       NOP
AFA3  EA       NOP
AFA4  EA       NOP
AFA5  EA       NOP
AFA6  A200     LDX  #$00
AFA8  A904     LDA  #$04
AFAA  8D1640   STA  $4016
AFAD  A00A     LDY  #$0A
AFAF  88       DEY
AFB0  D0FD     BNE  $AFAF
AFB2  EA       NOP
AFB3  EA       NOP
AFB4  AD1740   LDA  $4017
AFB7  4A       LSR
AFB8  290F     AND  #$0F
AFBA  9D3406   STA  $0634,X
AFBD  A906     LDA  #$06
AFBF  8D1640   STA  $4016
AFC2  A00A     LDY  #$0A
AFC4  88       DEY
AFC5  D0FD     BNE  $AFC4
AFC7  EA       NOP
AFC8  EA       NOP
AFC9  AD1740   LDA  $4017
AFCC  2A       ROL
AFCD  2A       ROL
AFCE  2A       ROL
AFCF  29F0     AND  #$F0
AFD1  1D3406   ORA  $0634,X
AFD4  49FF     EOR  #$FF
AFD6  A008     LDY  #$08
AFD8  0A       ASL
AFD9  7E3406   ROR  $0634,X
AFDC  88       DEY
AFDD  D0F9     BNE  $AFD8
AFDF  E8       INX
AFE0  E009     CPX  #$09
AFE2  D0C4     BNE  $AFA8
AFE4  A900     LDA  #$00
AFE6  856F     STA  $6F
AFE8  8570     STA  $70
AFEA  8571     STA  $71
AFEC  AD3B06   LDA  $063B
AFEF  2910     AND  #$10
AFF1  D02B     BNE  $B01E
AFF3  AD3B06   LDA  $063B
AFF6  2908     AND  #$08
AFF8  D044     BNE  $B03E
AFFA  AD3406   LDA  $0634
AFFD  2904     AND  #$04
AFFF  D045     BNE  $B046
B001  AD3B06   LDA  $063B
B004  2904     AND  #$04
B006  D04F     BNE  $B057
B008  202CB0   JSR  $B02C
B00B  D003     BNE  $B010
B00D  8518     STA  $18
B00F  60       RTS
B010  A418     LDY  $18
B012  D007     BNE  $B01B
B014  A450     LDY  $50
B016  D003     BNE  $B01B
B018  29FF     AND  #$FF
B01A  60       RTS
B01B  A900     LDA  #$00
B01D  60       RTS
```

键码翻译程序

```
B01E  A9FF     LDA  #$FF
B020  856F     STA  $6F
B022  8518     STA  $18
B024  AD3B06   LDA  $063B
B027  29E3     AND  #$E3
B029  8D3B06   STA  $063B
B02C  A200     LDX  #$00
B02E  A009     LDY  #$09
B030  BD3406   LDA  $0634,X
B033  D02B     BNE  $B060
```

```
B035  E8      INX
B036  88      DEY
B037  D0F7    BNE  $B030
B039  856D    STA  $6D
B03B  8550    STA  $50
B03D  60      RTS
B03E  A9FF    LDA  #$FF
B040  8518    STA  $18
B042  8570    STA  $70
B044  D0DE    BNE  $B024
B046  A9FF    LDA  #$FF
B048  8518    STA  $18
B04A  8570    STA  $70
B04C  AD3406  LDA  $0634
B04F  29FB    AND  #$FB
B051  8D3406  STA  $0634
B054  4C24B0  JMP  $B024
B057  A9FF    LDA  #$FF
B059  8571    STA  $71
B05B  8518    STA  $18
B05D  4C24B0  JMP  $B024
B060  866E    STX  $6E
B062  A000    LDY  #$00
B064  18      CLC
B065  4A      LSR
B066  B004    BCS  $B06C
B068  C8      INY
B069  4C65B0  JMP  $B065
B06C  8A      TXA
B06D  0A      ASL
B06E  0A      ASL
B06F  0A      ASL
B070  846D    STY  $6D
B072  18      CLC
B073  656D    ADC  $6D
B075  AA      TAX
B076  A56F    LDA  $6F
B078  D022    BNE  $B09C
B07A  A570    LDA  $70
B07C  D034    BNE  $B0B2
B07E  A571    LDA  $71
B080  D038    BNE  $B0BA
B082  A572    LDA  $72
B084  D03D    BNE  $B0C3
B086  BDECB0  LDA  $B0EC,X
B089  C9FF    CMP  #$FF
B08B  F003    BEQ  $B090
B08D  29FF    AND  #$FF
B08F  60      RTS
        大小写字母转换处理
B090  C550    CMP  $50
B092  F005    BEQ  $B099
B094  8550    STA  $50
;令$72=HFF,令光标为横杠
B096  20A1AD  JSR  $ADA1
B099  A900    LDA  #$00
B09B  60      RTS
B09C  BDECB0  LDA  $B0EC,X
B09F  C920    CMP  #$20
B0A1  B003    BCS  $B0A6
B0A3  29FF    AND  #$FF
B0A5  60      RTS
B0A6  38      SEC
B0A7  E940    SBC  #$40
B0A9  90EE    BCC  $B099
B0AB  C920    CMP  #$20
B0AD  B0EA    BCS  $B099
B0AF  29FF    AND  #$FF
B0B1  60      RTS
B0B2  A572    LDA  $72
B0B4  D013    BNE  $B0C9
B0B6  BD34B1  LDA  $B134,X
B0B9  60      RTS
;Alt 键的处理
B0BA  A572    LDA  $72
B0BC  D001    BNE  $B0BF
B0BE  60      RTS
B0BF  BD0CB2  LDA  $B20C,X
B0C2  60      RTS
;大、小写字母转换
B0C3  BD7CB1  LDA  $B17C,X
B0C6  4CCCB0  JMP  $B0CC
;改变光标形状
B0C9  BDC4B1  LDA  $B1C4,X
B0CC  C9FF    CMP  #$FF
B0CE  D0DF    BNE  $B0AF
B0D0  C550    CMP  $50
B0D2  F0C5    BEQ  $B099
```

```
B0D4   8550     STA   $50
B0D6   20A9AD   JSR   $ADA9
B0D9   A900     LDA   #$00
B0DB   60       RTS
B0DC   A008     LDY   #$08
B0DE   A200     LDX   #$00
B0E0   BD3406   LDA   $0634,X
B0E3   0A       ASL
B0E4   B0F5     BCS   $B0DB
B0E6   E8       INX
B0E7   88       DEY
B0E8   D0F6     BNE   $B0E0
B0EA   18       CLC
B0EB   60       RTS
```

键码数据区

```
B0EC                                                  03 7C 00 FF
B0F0   0B 5D 0D F7 3D 2D 2F 3D 3B 27 5B F6 12 50 2C 2E
B100   4B 4C 4F F5 38 39 4E 4D 4A 55 49 F4 36 37 56 42
B110   48 47 59 F3 34 35 43 46 44 52 54 F2 33 45 5A 58
B120   41 53 57 F1 32 31 00 00 00 51 1B F0 30 08 20 1F
B130   1C 1D 1E 09 03 5C 00 00 0C 7D 0D 18 2B 5F 3F 2B
B140   3A 22 7B 17 00 70 3C 3E 6B 6C 6F 11 2A 28 6E 6D
B150   6A 75 69 10 5E 26 76 62 68 67 79 0F 24 25 63 66
B160   64 72 74 0E 23 65 7A 78 61 73 77 06 40 21 00 00
B170   00 71 1B 02 29 08 20 1F 1C 1D 1E 0B 03 7C 00 FF
B180   0B 5D 0D F7 3D 2D 2F 3D 3B 27 5B F6 12 70 2C 2E
B190   6B 6C 6F F5 38 39 6E 6D 6A 75 69 F4 36 37 76 62
B1A0   68 67 79 F3 34 35 63 66 64 72 74 F2 33 65 7A 78
B1B0   61 73 77 F1 32 31 00 00 00 71 1B F0 30 08 20 1F
B1C0   1C 1D 1E 09 03 5C 00 00 0C 7D 0D F7 2B 5F 3F 2B
B1D0   3A 22 7B F6 12 50 3C 3E 4B 4C 4F F5 2A 28 4E 4D
B1E0   4A 55 49 F4 5E 26 56 42 48 47 59 F3 24 25 43 46
B1F0   44 52 54 F2 23 45 5A 58 41 53 57 F1 40 21 00 00
B200   00 51 1B F0 29 08 20 1F 1C 1D 1E 0B
```

MES.板制表图形数据区

```
B20C                                                  03 FE 00 00
B210   00 00 0D F7 FD F2 00 FD 00 00 00 F6 12 00 00 00
B220   00 00 00 F5 EF F0 00 00 00 00 00 F4 ED EE 00 00
B230   00 00 00 F3 EB EC 00 00 00 00 00 F2 EA 00 00 00
B240   00 00 00 F1 E9 E8 00 00 00 00 1B F0 F1 08 20 1F
B250   1C 1D 1E 0B
```

光标控制处理

```
B254   20A6AB   JSR   $ABA6
B257   A908     LDA   #$08
B259   D005     BNE   $B260
B25B   20A6AB   JSR   $ABA6
B25E   A909     LDA   #$09
B260   4C0888   JMP   $8808
```

读键处理

```
B263   A900     LDA   #$00
B265   855E     STA   $5E
B267   2088AF   JSR   $AF88
B26A   F046     BEQ   $B2B2
B26C   A666     LDX   $66
B26E   D025     BNE   $B295
B270   C575     CMP   $75
B272   D03E     BNE   $B2B2
B274   A57B     LDA   $7B
B276   D020     BNE   $B298
B278   2054B2   JSR   $B254
B27B   2088AF   JSR   $AF88
```

```
B27E    F02F      BEQ   $B2AF
B280    C575      CMP   $75
B282    D02B      BNE   $B2AF
B284    C677      DEC   $77
B286    D0F3      BNE   $B27B
B288    C678      DEC   $78
B28A    D0EF      BNE   $B27B
B28C    A9FF      LDA   #$FF
B28E    857B      STA   $7B
B290    4C06B3    JMP   $B306
B293    A903      LDA   #$03
B295    8575      STA   $75
B297    60        RTS
   连读键处理
B298    2088AF    JSR   $AF88
B29B    F015      BEQ   $B2B2
B29D    C575      CMP   $75
B29F    D011      BNE   $B2B2
B2A1    C679      DEC   $79
B2A3    D0F3      BNE   $B298
B2A5    A920      LDA   #$20
B2A7    8579      STA   $79
B2A9    2054B2    JSR   $B254
B2AC    4C06B3    JMP   $B306
;按键初期处理
B2AF    205BB2    JSR   $B25B
B2B2    A900      LDA   #$00
B2B4    857B      STA   $7B
B2B6    8577      STA   $77
B2B8    A920      LDA   #$20
B2BA    8579      STA   $79
B2BC    A901      LDA   #$01
B2BE    8578      STA   $78
;显示光标
B2C0    2054B2    JSR   $B254
B2C3    A980      LDA   #$80
B2C5    8574      STA   $74
;读键盘
B2C7    2088AF    JSR   $AF88
B2CA    D025      BNE   $B2F1
B2CC    C674      DEC   $74
B2CE    D0F7      BNE   $B2C7
B2D0    205BB2    JSR   $B25B
B2D3    A55E      LDA   $5E
B2D5    D0BC      BNE   $B293
B2D7    A980      LDA   #$80
B2D9    8574      STA   $74
B2DB    2088AF    JSR   $AF88
B2DE    D011      BNE   $B2F1
B2E0    C674      DEC   $74
B2E2    D0F7      BNE   $B2DB
B2E4    A55E      LDA   $5E
B2E6    D0AB      BNE   $B293
B2E8    4CB2B2    JMP   $B2B2
B2EB    205BB2    JSR   $B25B
B2EE    4CB2B2    JMP   $B2B2
B2F1    8575      STA   $75
B2F3    A566      LDA   $66
B2F5    D00F      BNE   $B306
B2F7    A901      LDA   #$01
B2F9    8574      STA   $74
B2FB    2088AF    JSR   $AF88
B2FE    C575      CMP   $75
B300    D0E9      BNE   $B2EB
B302    C674      DEC   $74
B304    D0F5      BNE   $B2FB
B306    205BB2    JSR   $B25B
;发按键声
B309    2014B3    JSR   $B314
B30C    A575      LDA   $75
B30E    60        RTS
   BEEP 保留字的入口
;发蜂鸣声
B30F    A9F0      LDA   #$F0
B311    8519      STA   $19
;发按键声
B313    AEA901    LDX   $01A9
B316    8519      STA   $19
B318    A901      LDA   #$01
B31A    8D1540    STA   $4015
B31D    A9C0      LDA   #$C0
B31F    8D1740    STA   $4017
B322    A9EF      LDA   #$EF
B324    8D0040    STA   $4000
B327    A900      LDA   #$00
B329    8D0140    STA   $4001
B32C    A95F      LDA   #$5F
B32E    8D0240    STA   $4002
```

```
B331   A000     LDY   #$00
B333   8C0340   STY   $4003
B336   88       DEY
B337   D0FD     BNE   $B336
B339   C619     DEC   $19
B33B   D0F9     BNE   $B336
B33D   A900     LDA   #$00
B33F   8D1540   STA   $4015
B342   60       RTS
```

读键主控程序

```
B343   A900     LDA   #$00
B345   8576     STA   $76
B347   8563     STA   $63
B349   2063B2   JSR   $B263
B34C   C90D     CMP   #$0D
B34E   F01E     BEQ   $B36E
B350   C903     CMP   #$03
B352   F00A     BEQ   $B35E
B354   C904     CMP   #$04
B356   F006     BEQ   $B35E
B358   200DAC   JSR   $AC0D
B35B   4C49B3   JMP   $B349
B35E   200DAC   JSR   $AC0D
B361   A676     LDX   $76
B363   A900     LDA   #$00
B365   9D0105   STA   $0501,X
B368   A903     LDA   #$03
B36A   9D0005   STA   $0500,X
B36D   60       RTS
```

语句的输入处理

```
B36E   A549     LDA   $49
B370   F00C     BEQ   $B37E
B372   AA       TAX
B373   CA       DEX
B374   BD0106   LDA   $0601,X
B377   F005     BEQ   $B37E
B379   C649     DEC   $49
B37B   4C6EB3   JMP   $B36E
B37E   A549     LDA   $49
B380   8547     STA   $47
B382   A902     LDA   #$02
B384   200888   JSR   $8808
B387   A564     LDA   $64
B389   D00A     BNE   $B395
B38B   A676     LDX   $76
B38D   CA       DEX
B38E   A900     LDA   #$00
B390   9D0005   STA   $0500,X
B393   8676     STX   $76
B395   A676     LDX   $76
B397   CA       DEX
B398   8A       TXA
B399   F007     BEQ   $B3A2
B39B   BD0005   LDA   $0500,X
B39E   C920     CMP   #$20
B3A0   F0F5     BEQ   $B397
B3A2   E8       INX
B3A3   A900     LDA   #$00
B3A5   9D0005   STA   $0500,X
B3A8   4C4EAC   JMP   $AC4E
```

FILE 中 SAVE、LOAD 指令辅助子程序系列

```
B3AB   E6AD     INC   $AD
B3AD   D002     BNE   $B3B1
B3AF   E6AE     INC   $AE
B3B1   60       RTS
B3B2   E691     INC   $91
B3B4   D0FB     BNE   $B3B1
B3B6   E692     INC   $92
B3B8   60       RTS
B3B9   A5AD     LDA   $AD
B3BB   C5AF     CMP   $AF
B3BD   D0F2     BNE   $B3B1
B3BF   A5AE     LDA   $AE
B3C1   C5B0     CMP   $B0
B3C3   60       RTS
B3C4   AD1405   LDA   $0514
B3C7   85AD     STA   $AD
B3C9   18       CLC
B3CA   6D1205   ADC   $0512
B3CD   85AF     STA   $AF
B3CF   AD1505   LDA   $0515
B3D2   85AE     STA   $AE
B3D4   6D1305   ADC   $0513
B3D7   85B0     STA   $B0
B3D9   60       RTS
B3DA   A900     LDA   #$00
B3DC   85AD     STA   $AD
B3DE   A905     LDA   #$05
```

```
B3E0    85AE     STA   $AE
B3E2    A980     LDA   #$80
B3E4    85AF     STA   $AF
B3E6    A905     LDA   #$05
B3E8    85B0     STA   $B0
B3EA    60       RTS
B3EB    A532     LDA   $32
B3ED    8D0020   STA   $2000
B3F0    60       RTS
B3F1    A532     LDA   $32
B3F3    0980     ORA   #$80
B3F5    8D0020   STA   $2000
B3F8    60       RTS
;FILE 中,执行 LOAD 命令的处理
B3F9    20EBB3   JSR   $B3EB
B3FC    A901     LDA   #$01
B3FE    4C06B4   JMP   $B406
B401    20EBB3   JSR   $B3EB
B404    A900     LDA   #$00
B406    85B1     STA   $B1
B408    A900     LDA   #$00
B40A    8592     STA   $92
B40C    20C4B3   JSR   $B3C4
B40F    A914     LDA   #$14
B411    8591     STA   $91
B413    20C0B4   JSR   $B4C0
B416    B00D     BCS   $B425
B418    D0F5     BNE   $B40F
B41A    C691     DEC   $91
B41C    D0F5     BNE   $B413
B41E    A914     LDA   #$14
B420    8591     STA   $91
B422    20C0B4   JSR   $B4C0
B425    B071     BCS   $B498
B427    F0E6     BEQ   $B40F
B429    C691     DEC   $91
B42B    D0F5     BNE   $B422
B42D    4C58B4   JMP   $B458
B430    20EBB3   JSR   $B3EB
B433    A900     LDA   #$00
B435    85B1     STA   $B1
B437    20DAB3   JSR   $B3DA
B43A    A928     LDA   #$28
B43C    8591     STA   $91
B43E    20C0B4   JSR   $B4C0
B441    B055     BCS   $B498
B443    D0F5     BNE   $B43A
B445    C691     DEC   $91
B447    D0F5     BNE   $B43E
B449    A928     LDA   #$28
B44B    8591     STA   $91
B44D    20C0B4   JSR   $B4C0
B450    B046     BCS   $B498
B452    F0E6     BEQ   $B43A
B454    C691     DEC   $91
B456    D0F5     BNE   $B44D
B458    A900     LDA   #$00
B45A    8591     STA   $91
B45C    8592     STA   $92
;读数送入五页
B45E    20A6B4   JSR   $B4A6
B461    20A6B4   JSR   $B4A6
B464    20D4B4   JSR   $B4D4
B467    B02F     BCS   $B498
B469    A6B1     LDX   $B1
B46B    D007     BNE   $B474
B46D    A000     LDY   #$00
B46F    91AD     STA   ($AD),Y
B471    4C7AB4   JMP   $B47A
B474    A000     LDY   #$00
B476    D1AD     CMP   ($AD),Y
B478    D01E     BNE   $B498
B47A    20ABB3   JSR   $B3AB
B47D    20B9B3   JSR   $B3B9
B480    D0E2     BNE   $B464
B482    A591     LDA   $91
B484    852C     STA   $2C
B486    A592     LDA   $92
B488    852D     STA   $2D
B48A    20D4B4   JSR   $B4D4
B48D    C52D     CMP   $2D
B48F    D007     BNE   $B498
B491    20D4B4   JSR   $B4D4
B494    C52C     CMP   $2C
B496    F007     BEQ   $B49F
B498    20F1B3   JSR   $B3F1
B49B    A902     LDA   #$02
B49D    38       SEC
```

```
B49E   60       RTS
B49F   20F1B3   JSR   $B3F1
B4A2   A900     LDA   #$00
B4A4   18       CLC
B4A5   60       RTS
;扫描 STOP 键
B4A6   20B986   JSR   $86B9
B4A9   F0F2     BEQ   $B49D
B4AB   AD1640   LDA   $4016
B4AE   2902     AND   #$02
B4B0   F0F4     BEQ   $B4A6
B4B2   20B986   JSR   $86B9
B4B5   F0E6     BEQ   $B49D
B4B7   AD1640   LDA   $4016
B4BA   2902     AND   #$02
B4BC   D0F4     BNE   $B4B2
B4BE   18       CLC
B4BF   60       RTS
;扫描 STOP 键
B4C0   20A6B4   JSR   $B4A6
B4C3   B0D8     BCS   $B49D
B4C5   A04E     LDY   #$4E
B4C7   8475     STY   $75
B4C9   C675     DEC   $75
B4CB   D0FC     BNE   $B4C9
B4CD   AD1640   LDA   $4016
B4D0   2902     AND   #$02
B4D2   18       CLC
B4D3   60       RTS
B4D4   A908     LDA   #$08
B4D6   8574     STA   $74
B4D8   A900     LDA   #$00
B4DA   8519     STA   $19
B4DC   A519     LDA   $19
B4DE   18       CLC
B4DF   6519     ADC   $19
B4E1   8519     STA   $19
B4E3   20C0B4   JSR   $B4C0
B4E6   B0B5     BCS   $B49D
B4E8   D005     BNE   $B4EF
B4EA   20B2B3   JSR   $B3B2
B4ED   E619     INC   $19
B4EF   C674     DEC   $74
B4F1   D0E9     BNE   $B4DC
;扫描 STOP 键
B4F3   20A6B4   JSR   $B4A6
B4F6   B0A5     BCS   $B49D
B4F8   A519     LDA   $19
B4FA   18       CLC
B4FB   60       RTS
;FILE 中,SAVE 时的处理
B4FC   20EBB3   JSR   $B3EB
B4FF   20DAB3   JSR   $B3DA
B502   A928     LDA   #$28
B504   8591     STA   $91
B506   A900     LDA   #$00
B508   8592     STA   $92
B50A   4C1BB5   JMP   $B51B
B50D   20EBB3   JSR   $B3EB
B510   20C4B3   JSR   $B3C4
B513   A914     LDA   #$14
B515   8591     STA   $91
B517   A900     LDA   #$00
B519   8592     STA   $92
B51B   A920     LDA   #$20
B51D   8519     STA   $19
B51F   A94E     LDA   #$4E
B521   851A     STA   $1A
B523   2086B5   JSR   $B586
B526   C619     DEC   $19
B528   D0F9     BNE   $B523
B52A   C61A     DEC   $1A
;执行 SAVE 时向录音机送数
B52C   D0F5     BNE   $B523
B52E   A591     LDA   $91
B530   8519     STA   $19
B532   2090B5   JSR   $B590
B535   C619     DEC   $19
B537   D0F9     BNE   $B532
B539   2086B5   JSR   $B586
B53C   C691     DEC   $91
B53E   D0F9     BNE   $B539
B540   2090B5   JSR   $B590
B543   A900     LDA   #$00
B545   8591     STA   $91
B547   8592     STA   $92
B549   A000     LDY   #$00
B54B   B1AD     LDA   ($AD),Y
```

```
B54D    206AB5    JSR    $B56A
B550    20ABB3    JSR    $B3AB
B553    20B9B3    JSR    $B3B9
B556    D0F1      BNE    $B549
B558    A591      LDA    $91
B55A    48        PHA
B55B    A592      LDA    $92
B55D    206AB5    JSR    $B56A
B560    68        PLA
B561    206AB5    JSR    $B56A
B564    2090B5    JSR    $B590
B567    4CF1B3    JMP    $B3F1
B56A    2090B5    JSR    $B590
B56D    A208      LDX    #$08
B56F    8519      STA    $19
B571    18        CLC
B572    6519      ADC    $19
B574    B007      BCS    $B57D
B576    2086B5    JSR    $B586
B579    CA        DEX
B57A    D0F3      BNE    $B56F
B57C    60        RTS
B57D    2090B5    JSR    $B590
B580    20B2B3    JSR    $B3B2
B583    4C79B5    JMP    $B579
;SAVE 中存图形时的处理
B586    48        PHA
B587    A934      LDA    #$34
B589    851B      STA    $1B
B58B    851C      STA    $1C
B58D    4C97B5    JMP    $B597
B590    48        PHA
B591    A96A      LDA    #$6A
B593    851B      STA    $1B
B595    851C      STA    $1C
B597    A904      LDA    #$04
B599    8D1640    STA    $4016
B59C    C61C      DEC    $1C
B59E    D0FC      BNE    $B59C
B5A0    A9FF      LDA    #$FF
B5A2    8D1640    STA    $4016
B5A5    C61B      DEC    $1B
B5A7    D0FC      BNE    $B5A5
B5A9    68        PLA
B5AA    60        RTS
      BG GRAPHIC 画面的绘制及工作
      状态的建立
B5AB    A900      LDA    #$00
B5AD    859E      STA    $9E
B5AF    20DFA6    JSR    $A6DF
B5B2    2082AA    JSR    $AA82
B5B5    A59E      LDA    $9E
B5B7    C981      CMP    #$81
B5B9    F003      BEQ    $B5BE
;清背景一页
B5BB    2071AB    JSR    $AB71
B5BE    20FEAA    JSR    $AAFE
B5C1    20E6AB    JSR    $ABE6
B5C4    A532      LDA    $32
B5C6    0908      ORA    #$08
B5C8    8532      STA    $32
B5CA    A924      LDA    #$24
B5CC    854F      STA    $4F
B5CE    20F1B3    JSR    $B3F1
B5D1    20C0A7    JSR    $A7C0
B5D4    A900      LDA    #$00
B5D6    8D0505    STA    $0505
B5D9    8D0605    STA    $0506
B5DC    8588      STA    $88
B5DE    8D0405    STA    $0504
B5E1    4C82B8    JMP    $B882
;定义框标
B5E4    AD0505    LDA    $0505
B5E7    8548      STA    $48
B5E9    AD0605    LDA    $0506
B5EC    8549      STA    $49
B5EE    60        RTS
B5EF    85E0      STA    $E0
B5F1    85E1      STA    $E1
B5F3    20A9AD    JSR    $ADA9
B5F6    4C63B2    JMP    $B263
      BG GRAPHIC 中按方向键的处理
B5F9    C91C      CMP    #$1C
B5FB    F00D      BEQ    $B60A
B5FD    C91D      CMP    #$1D
B5FF    F016      BEQ    $B617
B601    C91E      CMP    #$1E
B603    F022      BEQ    $B627
```

```
B605    C91F      CMP    #$1F
B607    F02E      BEQ    $B637
B609    60        RTS
```

;按 → 键的处理

```
B60A    AD0505    LDA    $0505
B60D    C91B      CMP    #$1B
B60F    F021      BEQ    $B632
B611    EE0505    INC    $0505
B614    A900      LDA    #$00
B616    60        RTS
```

;按 ← 键的处理

```
B617    AD0505    LDA    $0505
B61A    F006      BEQ    $B622
B61C    CE0505    DEC    $0505
B61F    4C14B6    JMP    $B614
B622    A91B      LDA    #$1B
B624    8D0505    STA    $0505
```

;按 ↑ 键的处理

```
B627    AD0605    LDA    $0506
B62A    F0E8      BEQ    $B614
B62C    CE0605    DEC    $0506
B62F    4C14B6    JMP    $B614
B632    A900      LDA    #$00
B634    8D0505    STA    $0505
```

;按 ↓ 键的处理

```
B637    AD0605    LDA    $0506
B63A    C914      CMP    #$14
B63C    F0D6      BEQ    $B614
B63E    EE0605    INC    $0506
B641    D0D1      BNE    $B614
```

;BG GRAPHIC 中按 ESC 键的处理

```
B643    20CCA7    JSR    $A7CC
B646    20B5B6    JSR    $B6B5
B649    208BAD    JSR    $AD8B
B64C    2001B8    JSR    $B801
B64F    208BAD    JSR    $AD8B
```

;BG GRAPHIC 指令的判断处理

```
B652    A93E      LDA    #$3E
B654    20EFB5    JSR    $B5EF
B657    C91E      CMP    #$1E
B659    F024      BEQ    $B67F
B65B    C91F      CMP    #$1F
B65D    F029      BEQ    $B688
B65F    C920      CMP    #$20
B661    F02F      BEQ    $B692
B663    C903      CMP    #$03
B665    F002      BEQ    $B669
B667    D0E9      BNE    $B652
```

;按 STOP 键的处理

```
B669    20B8B6    JSR    $B6B8
B66C    208FAD    JSR    $AD8F
B66F    A915      LDA    #$15
B671    8549      STA    $49
B673    A91A      LDA    #$1A
B675    200DAC    JSR    $AC0D
B678    A981      LDA    #$81
B67A    859E      STA    $9E
B67C    4CAD80    JMP    $80AD
B67F    A549      LDA    $49
B681    F0CF      BEQ    $B652
B683    C649      DEC    $49
B685    4C52B6    JMP    $B652
B688    A549      LDA    $49
B68A    C905      CMP    #$05
B68C    F0C4      BEQ    $B652
B68E    E649      INC    $49
B690    D0C0      BNE    $B652
B692    A549      LDA    $49
B694    0A        ASL
B695    AA        TAX
B696    BDA3B6    LDA    $B6A3,X
B699    8519      STA    $19
B69B    BDA4B6    LDA    $B6A4,X
B69E    851A      STA    $1A
B6A0    6C1900    JMP    ($0019)
```

BG GRAPHIC 指令入口数据区

```
B6A3        7F B8    22 BB   1F BB   AB B5   FB BB   2D B7
            SELECT   COPY    MOVE    CLEAR   FILE    CHAR
```

BG GRAPHIC 中按 ESC 键的处理

```
B6AF    20E6AB    JSR    $ABE6
B6B2    4CFAAB    JMP    $ABFA
B6B5    A9FF      LDA    #$FF
B6B7    AEA900    LDX    $00A9
```

```
B6BA    8D0005   STA    $0500
B6BD    20F1AB   JSR    $ABF1
B6C0    A909     LDA    #$09
B6C2    8574     STA    $74
B6C4    A900     LDA    #$00
B6C6    853D     STA    $3D
B6C8    208BAD   JSR    $AD8B
B6CB    20A6AB   JSR    $ABA6
B6CE    20D8AB   JSR    $ABD8
;在画面左上角挖出一块空地显示指令
B6D1    AD0005   LDA    $0500
B6D4    F003     BEQ    $B6D9
B6D6    AD0720   LDA    $2007
B6D9    A00A     LDY    #$0A
B6DB    A63D     LDX    $3D
B6DD    AD0005   LDA    $0500
B6E0    F009     BEQ    $B6EB
B6E2    AD0720   LDA    $2007
B6E5    9D2005   STA    $0520,X
B6E8    4CF1B6   JMP    $B6F1
;恢复原来的图形画面         B6EB
BD2005  LDA     $0520,X
B6EE    8D0720   STA    $2007
B6F1    E8       INX
B6F2    88       DEY
B6F3    D0E8     BNE    $B6DD
B6F5    863D     STX    $3D
B6F7    C674     DEC    $74
B6F9    F0B4     BEQ    $B6AF
B6FB    E649     INC    $49
B6FD    D0CC     BNE    $B6CB
    定义 BG GRAPHIC 画面的框标
;清卡通定义区
B6FF    20DFA6   JSR    $A6DF
B702    A9B0     LDA    #$B0
B704    8D0102   STA    $0201
B707    A900     LDA    #$00
B709    8D0202   STA    $0202
B70C    AD0505   LDA    $0505
B70F    18       CLC
B710    6902     ADC    #$02
B712    0A       ASL
B713    0A       ASL
B714    0A       ASL
B715    8D0302   STA    $0203
B718    AD0605   LDA    $0506
B71B    18       CLC
B71C    6903     ADC    #$03
B71E    0A       ASL
B71F    0A       ASL
B720    0A       ASL
B721    8D0002   STA    $0200
B724    CE0002   DEC    $0200
B727    4C83B7   JMP    $B783
B72A    4C43B6   JMP    $B643
    BG GRAPHIC——CHAR 指令的处理
B72D    20F3AA   JSR    $AAF3
B730    20B8B6   JSR    $B6B8
B733    A905     LDA    #$05
B735    201BB8   JSR    $B81B
B738    20CCA7   JSR    $A7CC
B73B    20E4B5   JSR    $B5E4
B73E    20A1AC   JSR    $ACA1
B741    A548     LDA    $48
B743    8D0505   STA    $0505
B746    A549     LDA    $49
B748    8D0605   STA    $0506
B74B    2083B7   JSR    $B783
B74E    20AAAC   JSR    $ACAA
B751    2063B2   JSR    $B263
B754    C90D     CMP    #$0D
B756    F00F     BEQ    $B767
B758    C91B     CMP    #$1B
B75A    F0CE     BEQ    $B72A
B75C    90E0     BCC    $B73E
B75E    200DAC   JSR    $AC0D
B761    2070B7   JSR    $B770
B764    4C3EB7   JMP    $B73E
B767    2007AC   JSR    $AC07
B76A    2070B7   JSR    $B770
B76D    4C3EB7   JMP    $B73E
B770    A549     LDA    $49
B772    C915     CMP    #$15
B774    B001     BCS    $B777
B776    60       RTS
B777    A914     LDA    #$14
B779    8549     STA    $49
B77B    60       RTS
```

```
B77C    58 3A 0D 0A 59 3A 00
```

BG GRAPHIC 状态的建立和工作处理程序

```
B783    208FAD    JSR    $AD8F
B786    A916      LDA    #$16
B788    8549      STA    $49
B78A    A97C      LDA    #$7C
B78C    8552      STA    $52
B78E    A9B7      LDA    #$B7
B790    8553      STA    $53
B792    20F887    JSR    $87F8
B795    AD0505    LDA    $0505
B798    8528      STA    $28
B79A    A902      LDA    #$02
B79C    8548      STA    $48
B79E    A916      LDA    #$16
B7A0    8549      STA    $49
B7A2    20B0B7    JSR    $B7B0
B7A5    AD0605    LDA    $0506
B7A8    8528      STA    $28
B7AA    A902      LDA    #$02
B7AC    8548      STA    $48
B7AE    E649      INC    $49
B7B0    20CDA1    JSR    $A1CD
B7B3    203C8D    JSR    $8D3C
B7B6    A93D      LDA    #$3D
B7B8    18        CLC
B7B9    6903      ADC    #$03
B7BB    8552      STA    $52
B7BD    A900      LDA    #$00
B7BF    8553      STA    $53
B7C1    4CF887    JMP    $87F8
```

BG GRAPHIC 指令数据区

```
B7C4            4D 4F 44 45 20 20 00 53 45 4C 45 43
                MODE                 SELECT
B7D0  54 00 4D 4F 56 45 20 20 00 43 4F 50 59 20 20 00
            MOVE                     COPY
B7E0  43 48 41 52 20 20 00 43 4C 45 41 52 20 00 46 49
      CHAR                  CLEAR                FILE
B7F0  4C 45 20 20 00 CB B7  D9 B7  D2 B7  E7 B7  EE B7  E0 B7
      指令数据区入口   SELECT COPY   MOVE   CLEAR  FILE   CHAR
```

BG GRAPHIC 中按 ESC 键后送显各指令的处理

```
B801    A900      LDA    #$00
B803    8575      STA    $75
B805    A920      LDA    #$20
B807    20F087    JSR    $87F0
B80A    A575      LDA    $75
B80C    2038B8    JSR    $B838
B80F    2007AC    JSR    $AC07
B812    E675      INC    $75
B814    A575      LDA    $75
B816    C906      CMP    #$06
B818    D0EB      BNE    $B805
B81A    60        RTS
```

BG GRAPHIC 画面的绘制

```
B81B    857C      STA    $7C
B81D    A905      LDA    #$05
B81F    8548      STA    $48
B821    A916      LDA    #$16
B823    8549      STA    $49
B825    A9C4      LDA    #$C4
B827    8552      STA    $52
B829    A9B7      LDA    #$B7
B82B    8553      STA    $53
B82D    20F887    JSR    $87F8
B830    A905      LDA    #$05
B832    8548      STA    $48
B834    E649      INC    $49
B836    A57C      LDA    $7C
B838    0A        ASL
B839    AA        TAX
B83A    BDF5B7    LDA    $B7F5,X
B83D    8552      STA    $52
B83F    BDF6B7    LDA    $B7F6,X
B842    8553      STA    $53
B844    4CF887    JMP    $87F8
```

背景图形库中每组图形的起始代码

```
B847    00 08 10 18 B8 C0 C8 D0 D8 E0
        E8 F0 F8
```

BG GRAPHIC 板按回车键的处理

```
B854    205AB8    JSR    $B85A
B857    4C9DB8    JMP    $B89D
B85A    20EDB8    JSR    $B8ED
B85D    20FEAA    JSR    $AAFE
B860    A927      LDA    #$27
B862    A22C      LDX    #$2C
B864    20F881    JSR    $81F8
B867    A588      LDA    $88
B869    0930      ORA    #$30
B86B    4CE3AB    JMP    $ABE3
```

BG GRAPHIC 板按 D 键的删除处理

```
B86E    20A1AC    JSR    $ACA1
B871    20E4B5    JSR    $B5E4
B874    A920      LDA    #$20
B876    200DAC    JSR    $AC0D
B879    20AAAC    JSR    $ACAA
B87C    4C9DB8    JMP    $B89D
B87F    20B8B6    JSR    $B6B8
B882    20C0A7    JSR    $A7C0
B885    A900      LDA    #$00
B887    857F      STA    $7F
B889    857D      STA    $7D
B88B    201BB8    JSR    $B81B
B88E    A903      LDA    #$03
B890    8588      STA    $88
B892    205AB8    JSR    $B85A
B895    A90C      LDA    #$0C
B897    8548      STA    $48
B899    A917      LDA    #$17
B89B    8549      STA    $49
B89D    20A2B9    JSR    $B9A2
B8A0    20A1AC    JSR    $ACA1
B8A3    20FFB6    JSR    $B6FF
B8A6    20AAAC    JSR    $ACAA
B8A9    A95E      LDA    #$5E
```

;设置图形指示器

```
B8AB    20EFB5    JSR    $B5EF
```

;检查按键是否合法·

```
B8AE    20F9B5    JSR    $B5F9
B8B1    A200      LDX    #$00
B8B3    DDCFB8    CMP    $B8CF,X
B8B6    F007      BEQ    $B8BF
B8B8    E8        INX
B8B9    E008      CPX    #$08
B8BB    D0F6      BNE    $B8B3
B8BD    F0E1      BEQ    $B8A0
```

;查到合法键值的处理

```
B8BF    8A        TXA
B8C0    0A        ASL
B8C1    AA        TAX
B8C2    BDD7B8    LDA    $B8D7,X
B8C5    8519      STA    $19
B8C7    BDD8B8    LDA    $B8D8,X
B8CA    851A      STA    $1A
B8CC    6C1900    JMP    ($0019)
```

BG GRAPHIC 键码数据区

```
B8CF                                                1B
B8D0    0D 0C 0B 08 12 20 44 17 B9 54 B8 2E B9 37 B9 41
B8E0    B9 4F B9 5D B9 6E B8
```

BG GRAPHIC 板 PPU 配色处理

```
B8E7    A900      LDA    #$00
B8E9    8588      STA    $88
B8EB    F008      BEQ    $B8F5
B8ED    E688      INC    $88
B8EF    A588      LDA    $88
B8F1    C904      CMP    #$04
B8F3    F0F2      BEQ    $B8E7
B8F5    0A        ASL
B8F6    0A        ASL
B8F7    0588      ORA    $88
B8F9    48        PHA
B8FA    20FEAA    JSR    $AAFE
B8FD    A927      LDA    #$27
B8FF    A2F3      LDX    #$F3
B901    20F881    JSR    $81F8
B904    68        PLA
B905    AA        TAX
B906    29FC      AND    #$FC
B908    8D0720    STA    $2007
B90B    8A        TXA
```

```
B90C    A204      LDX    #$04
B90E    8D0720    STA    $2007
B911    CA        DEX
B912    D0FA      BNE    $B90E
B914    4CE6AB    JMP    $ABE6
```

BG GRAPHIC 按 ESC 键的处理

```
B917    A90C      LDA    #$0C
B919    8548      STA    $48
B91B    A916      LDA    #$16
B91D    8549      STA    $49
B91F    A905      LDA    #$05
;清图形提示行中的图形
B921    200DAC    JSR    $AC0D
B924    E649      INC    $49
B926    A905      LDA    #$05
;清图形提示行中的光标
B928    200DAC    JSR    $AC0D
B92B    4C43B6    JMP    $B643
```

BG GRAPHIC 按 Home+Shift 键的处理

```
B92E    A57D      LDA    $7D
B930    F002      BEQ    $B934
B932    C67D      DEC    $7D
B934    4C9DB8    JMP    $B89D
```

BG GRAPHIC 按 Home 键的处理

```
B937    A57D      LDA    $7D
B939    C90C      CMP    #$0C
B93B    F0F7      BEQ    $B934
B93D    E67D      INC    $7D
B93F    D0F3      BNE    $B934
;按 Del 键的处理
B941    A548      LDA    $48
B943    C91A      CMP    #$1A
B945    F0ED      BEQ    $B934
B947    E648      INC    $48
B949    E648      INC    $48
B94B    E67F      INC    $7F
B94D    D0E5      BNE    $B934
;按 Ins 键的处理
B94F    A548      LDA    $48
B951    C90C      CMP    #$0C
B953    F0DF      BEQ    $B934
B955    C648      DEC    $48
B957    C648      DEC    $48
B959    C67F      DEC    $7F
B95B    10D7      BPL    $B934
;按空格键的处理
B95D    20A1AC    JSR    $ACA1
B960    20E4B5    JSR    $B5E4
B963    20A6AB    JSR    $ABA6
B966    A67D      LDX    $7D
B968    BD47B8    LDA    $B847,X
B96B    18        CLC
B96C    657F      ADC    $7F
B96E    8573      STA    $73
B970    A909      LDA    #$09
B972    200888    JSR    $8808
B975    A588      LDA    $88
B977    859C      STA    $9C
B979    2067BA    JSR    $BA67
B97C    EE0505    INC    $0505
B97F    AD0505    LDA    $0505
B982    C91C      CMP    #$1C
B984    F006      BEQ    $B98C
B986    20AAAC    JSR    $ACAA
B989    4C9DB8    JMP    $B89D
```

BG GRAPHIC 输入图形达到右边界的处理

```
B98C    A900      LDA    #$00
B98E    8D0505    STA    $0505
B991    EE0605    INC    $0506
B994    AD0605    LDA    $0506
B997    C915      CMP    #$15
B999    D0EB      BNE    $B986
B99B    A914      LDA    #$14
B99D    8D0605    STA    $0506
B9A0    10E4      BPL    $B986
;送显八个图形样板
B9A2    20A1AC    JSR    $ACA1
B9A5    A90C      LDA    #$0C
B9A7    8548      STA    $48
B9A9    A916      LDA    #$16
B9AB    8549      STA    $49
B9AD    A67D      LDX    $7D
B9AF    BD47B8    LDA    $B847,X
B9B2    857E      STA    $7E
B9B4    A008      LDY    #$08
B9B6    A200      LDX    #$00
B9B8    A57E      LDA    $7E
```

```
B9BA   9D1005   STA   $0510,X
B9BD   E8       INX
B9BE   A920     LDA   #$20
B9C0   9D1005   STA   $0510,X
B9C3   E8       INX
B9C4   E67E     INC   $7E
B9C6   88       DEY
B9C7   D0EF     BNE   $B9B8
B9C9   20A6AB   JSR   $ABA6
B9CC   A200     LDX   #$00
B9CE   A008     LDY   #$08
B9D0   20FEAA   JSR   $AAFE
B9D3   20D8AB   JSR   $ABD8
B9D6   BD1005   LDA   $0510,X
B9D9   8D0720   STA   $2007
B9DC   E8       INX
B9DD   A920     LDA   #$20
B9DF   8D0720   STA   $2007
B9E2   E8       INX
B9E3   88       DEY
B9E4   D0F0     BNE   $B9D6
B9E6   20E6AB   JSR   $ABE6
B9E9   4CAAAC   JMP   $ACAA
```

BG GRAPHIC 板改变图形配色

```
B9EC   E648     INC   $48
B9EE   E648     INC   $48
B9F0   A549     LDA   $49
B9F2   18       CLC
B9F3   6903     ADC   #$03
B9F5   8549     STA   $49
B9F7   29FC     AND   #$FC
B9F9   0A       ASL
B9FA   854A     STA   $4A
B9FC   A548     LDA   $48
B9FE   4A       LSR
B9FF   4A       LSR
BA00   18       CLC
BA01   654A     ADC   $4A
BA03   69C0     ADC   #$C0
BA05   854A     STA   $4A
BA07   A54F     LDA   $4F
BA09   6903     ADC   #$03
BA0B   854B     STA   $4B
BA0D   A902     LDA   #$02
BA0F   2549     AND   $49
BA11   8519     STA   $19
BA13   A548     LDA   $48
BA15   2903     AND   #$03
BA17   4A       LSR
BA18   18       CLC
BA19   6519     ADC   $19
BA1B   8519     STA   $19
BA1D   A9FC     LDA   #$FC
BA1F   851A     STA   $1A
BA21   0619     ASL   $19
BA23   A619     LDX   $19
BA25   F008     BEQ   $BA2F
BA27   069C     ASL   $9C
BA29   38       SEC
BA2A   261A     ROL   $1A
BA2C   CA       DEX
BA2D   D0F8     BNE   $BA27
BA2F   A905     LDA   #$05
BA31   200888   JSR   $8808
BA34   A573     LDA   $73
BA36   60       RTS
```

BG GRAPHIC 中 COPY、MOVE 指令的处理

```
BA37   20ECB9   JSR   $B9EC
BA3A   859C     STA   $9C
BA3C   A51A     LDA   $1A
BA3E   49FF     EOR   #$FF
BA40   259C     AND   $9C
BA42   859C     STA   $9C
BA44   A419     LDY   $19
BA46   F005     BEQ   $BA4D
BA48   469C     LSR   $9C
BA4A   88       DEY
BA4B   D0FB     BNE   $BA48
BA4D   60       RTS
```

COLOR 保留字的入口

```
;读 X 座标值
BA4E   206D9C   JSR   $9C6D
BA51   48       PHA
;读 Y 座标值
BA52   206A9C   JSR   $9C6A
BA55   AD0004   LDA   $0400
BA58   8549     STA   $49
```

```
BA5A   68       PLA
BA5B   8548     STA   $48
;读配色代码
BA5D   206A9C   JSR   $9C6A
BA60   AD0004   LDA   $0400
BA63   2903     AND   #$03
BA65   859C     STA   $9C
BA67   20ECB9   JSR   $B9EC
BA6A   251A     AND   $1A
BA6C   059C     ORA   $9C
BA6E   8573     STA   $73
BA70   A909     LDA   #$09
BA72   4C0888   JMP   $8808
```

BG GRAPHIC 中 SAVE 指令的处理

```
BA75   A903     LDA   #$03
BA77   8D0005   STA   $0500
BA7A   A900     LDA   #$00
BA7C   8D1405   STA   $0514
BA7F   8D1605   STA   $0516
BA82   A907     LDA   #$07
BA84   8D1505   STA   $0515
BA87   8D1705   STA   $0517
BA8A   A200     LDX   #$00
BA8C   8E1205   STX   $0512
BA8F   E8       INX
BA90   8E1305   STX   $0513
BA93   A900     LDA   #$00
BA95   855F     STA   $5F
BA97   A924     LDA   #$24
BA99   8560     STA   $60
BA9B   20C397   JSR   $97C3
BA9E   20F1AB   JSR   $ABF1
BAA1   A560     LDA   $60
BAA3   A65F     LDX   $5F
BAA5   20F881   JSR   $81F8
BAA8   AD0720   LDA   $2007
BAAB   A200     LDX   #$00
BAAD   AD0720   LDA   $2007
BAB0   9D0007   STA   $0700,X
BAB3   E8       INX
BAB4   D0F7     BNE   $BAAD
BAB6   20E6AB   JSR   $ABE6
BAB9   20FAAB   JSR   $ABFA
BABC   20FCB4   JSR   $B4FC
BABF   200DB5   JSR   $B50D
BAC2   E660     INC   $60
BAC4   A560     LDA   $60
BAC6   C928     CMP   #$28
BAC8   D0D4     BNE   $BA9E
BACA   60       RTS
```

FILE 中执行 LOAD 的处理

```
BACB   20DD97   JSR   $97DD
BACE   A900     LDA   #$00
BAD0   855F     STA   $5F
BAD2   A924     LDA   #$24
BAD4   8560     STA   $60
BAD6   2030B4   JSR   $B430
BAD9   B0EF     BCS   $BACA
BADB   AD0005   LDA   $0500
BADE   C903     CMP   #$03
BAE0   D0E9     BNE   $BACB
;核对文件名
BAE2   20C898   JSR   $98C8
BAE5   B0E4     BCS   $BACB
;送显 LOADING 文件名
BAE7   20D097   JSR   $97D0
BAEA   2001B4   JSR   $B401
BAED   B0DB     BCS   $BACA
BAEF   20F1AB   JSR   $ABF1
BAF2   A560     LDA   $60
BAF4   A65F     LDX   $5F
BAF6   20F881   JSR   $81F8
BAF9   A200     LDX   #$00
;把七页的数据送入 PPU
BAFB   BD0007   LDA   $0700,X
BAFE   8D0720   STA   $2007
BB01   E8       INX
BB02   D0F7     BNE   $BAFB
BB04   20E6AB   JSR   $ABE6
BB07   20FAAB   JSR   $ABFA
BB0A   E660     INC   $60
BB0C   A560     LDA   $60
BB0E   C928     CMP   #$28
BB10   D0C4     BNE   $BAD6
BB12   60       RTS
```

BG GRAPHIC——MOVE、COPY 指令的处理

```
BB13   A900     LDA   #$00
```

```
BB15    85AE      STA    $AE
BB17    A9F8      LDA    #$F8
BB19    8D1002    STA    $0210
BB1C    4C43B6    JMP    $B643
 BG GRAPHIC——MOVE 指令的处理
BB1F    A902      LDA    #$02
BB21    AEA901    LDX    $01A9
BB24    857C      STA    $7C
BB26    AA        TAX
BB27    CA        DEX
BB28    86AF      STX    $AF
BB2A    20B8B6    JSR    $B6B8
BB2D    20C0A7    JSR    $A7C0
BB30    A900      LDA    #$00
BB32    85AE      STA    $AE
BB34    201DB8    JSR    $B81D
BB37    20FFB6    JSR    $B6FF
BB3A    20ADBB    JSR    $BBAD
BB3D    A920      LDA    #$20
BB3F    20EFB5    JSR    $B5EF
BB42    20F9B5    JSR    $B5F9
BB45    C91B      CMP    #$1B
;是 Esc 转 $BB13
BB47    F0CA      BEQ    $BB13
BB49    C912      CMP    #$12
;是 Ins 转 $BB7D
BB4B    F030      BEQ    $BB7D
BB4D    C908      CMP    #$08
;是 Del 转 $BB53
BB4F    F002      BEQ    $BB53
BB51    D0E4      BNE    $BB37
;BG GRAPHIC——MOVE 或
COPY 指令中按 Del 键的处理
BB53    A5AE      LDA    $AE
BB55    F0E0      BEQ    $BB37
BB57    20A1AC    JSR    $ACA1
BB5A    20E4B5    JSR    $B5E4
BB5D    20A6AB    JSR    $ABA6
BB60    A591      LDA    $91
BB62    8573      STA    $73
BB64    A909      LDA    #$09
BB66    200888    JSR    $8808
BB69    A588      LDA    $88
BB6B    859C      STA    $9C
BB6D    2067BA    JSR    $BA67
BB70    20AAAC    JSR    $ACAA
BB73    A5AF      LDA    $AF
BB75    F004      BEQ    $BB7B
BB77    A900      LDA    #$00
BB79    85AE      STA    $AE
BB7B    F0BA      BEQ    $BB37
;COPY、MOVE 指令中按 Ins 键的处理
BB7D    20A1AC    JSR    $ACA1
BB80    20E4B5    JSR    $B5E4
BB83    20A6AB    JSR    $ABA6
BB86    A905      LDA    #$05
BB88    200888    JSR    $8808
BB8B    A573      LDA    $73
BB8D    8591      STA    $91
BB8F    A5AF      LDA    $AF
BB91    F009      BEQ    $BB9C
BB93    A920      LDA    #$20
BB95    8573      STA    $73
BB97    A909      LDA    #$09
BB99    200888    JSR    $8808
BB9C    A9FF      LDA    #$FF
BB9E    85AE      STA    $AE
BBA0    2037BA    JSR    $BA37
BBA3    A59C      LDA    $9C
BBA5    8588      STA    $88
BBA7    20AAAC    JSR    $ACAA
BBAA    4C37BB    JMP    $BB37
;COPY、MOVE 指令执行中对图形的定义处理
BBAD    A5AE      LDA    $AE
BBAF    F028      BEQ    $BBD9
BBB1    A591      LDA    $91
BBB3    8D1102    STA    $0211
BBB6    A588      LDA    $88
BBB8    290F      AND    #$0F
BBBA    8D1202    STA    $0212
BBBD    AD0505    LDA    $0505
BBC0    18        CLC
BBC1    6902      ADC    #$02
BBC3    0A        ASL
BBC4    0A        ASL
BBC5    0A        ASL
BBC6    8D1302    STA    $0213
BBC9    AD0605    LDA    $0506
```

```
BBCC   18       CLC
BBCD   6903     ADC    #$03
BBCF   0A       ASL
BBD0   0A       ASL
BBD1   0A       ASL
BBD2   8D1002   STA    $0210
BBD5   CE1002   DEC    $0210
BBD8   60       RTS
```

;BG GRAPHIC——COPY

指令中解除认定图形的处理

```
BBD9   A9F8     LDA    #$F8
BBDB   8D1002   STA    $0210
BBDE   60       RTS
```

```
BBDF   53 41 56 45 28 53 29 2C 4C 4F 41 44 28 4C 29 3F 00
        SAVE(S),              LOAD(L)?
BBF0   46 49 4C 45 20 4E 41 4D 45 22 00
          FILE NAME"
```

BG———FILE 指令处理

```
BBFB   20B8B6   JSR    $B6B8
BBFE   208FAD   JSR    $AD8F
BC01   A916     LDA    #$16
BC03   8549     STA    $49
BC05   A91A     LDA    #$1A
BC07   200DAC   JSR    $AC0D
BC0A   A920     LDA    #$20
BC0C   854F     STA    $4F
BC0E   20F1B3   JSR    $B3F1
BC11   20F3AA   JSR    $AAF3
BC14   2075AB   JSR    $AB75
BC17   A9DF     LDA    #$DF
BC19   8552     STA    $52
BC1B   A9BB     LDA    #$BB
BC1D   8553     STA    $53
```

;送显 SAVE(S),LOAD(L)?

建立 FILE 工作状态

```
BC1F   20F887   JSR    $87F8
BC22   2063B2   JSR    $B263
BC25   C953     CMP    #$53
BC27   F006     BEQ    $BC2F
BC29   C94C     CMP    #$4C
BC2B   F012     BEQ    $BC3F
BC2D   D009     BNE    $BC38
```

;按"S"保存图形

```
BC2F   200DAC   JSR    $AC0D
BC32   204BBC   JSR    $BC4B
BC35   2075BA   JSR    $BA75
BC38   A981     LDA    #$81
BC3A   859E     STA    $9E
BC3C   4CAFB5   JMP    $B5AF
```

;按"L"读回图形

```
BC3F   200DAC   JSR    $AC0D
BC42   204BBC   JSR    $BC4B
BC45   20CEBA   JSR    $BACE
BC48   4C38BC   JMP    $BC38
```

FILE 子程序，送显 FILE NAME

```
BC4B   2007AC   JSR    $AC07
BC4E   A9F0     LDA    #$F0
BC50   8552     STA    $52
BC52   A9BB     LDA    #$BB
BC54   8553     STA    $53
BC56   20F887   JSR    $87F8
BC59   2043B3   JSR    $B343
BC5C   AD0005   LDA    $0500
BC5F   C903     CMP    #$03
BC61   F00D     BEQ    $BC70
BC63   20328A   JSR    $8A32
BC66   A909     LDA    #$09
BC68   8509     STA    $09
```

;把文件名存入五页

```
BC6A   206998   JSR    $9869
BC6D   4CB498   JMP    $98B4
BC70   68       PLA
BC71   68       PLA
BC72   4C38BC   JMP    $BC38
```

十六类运动卡通的取数地址

```
BC75            A5 BC  A5 BC  B1 BC  B7 BC  BD BC  AB
                 0#     1#     2#     3#     4#    5#
```

```
BC80    BC B7 BC B1 BC C3 BC C3 BC B1 BC B1 BC C9 BC AB
          6#      7#    8#     9#     10#    11#    12#    13#
BC90    BC B1 BC AB BC 00 1C 00 00 00 00 38 58 00 0C 78
          14#  15#    16#
BCA0    20 00 48 88 58 F3 BC 29 BD CF BC FC BC 95 BD EA
BCB0    BC FC BC 3B BD D8 BC FC BC 4D BD E1 BC 0E BD 5F
BCC0    BD EA BC 05 BD 71 BD E1 BC 05 BD 83 BD E1 BC 00
BCD0    00 40 40 40 00 00 00 00 00 00 00 00 00 00 00 00
BCE0    00 00 00 40 40 C0 80 80 00 00 00 00 40 40 40 00
BCF0    00 00 00 10 20 10 30 10 20 10 30 10 20 20 20 20
BD00    20 20 20 20 20 10 10 10 10 10 10 10 10 10 10 20
BD10    20 20 20 20 20 20 20 00 00 00 FE 02 FE 02 00 02
BD20    02 00 02 FE 02 FE 00 FE FE C7 BD C3 BD CF BD BF
BD30    BD CF BD C3 BD CB BD BB BD CB BD D7 BD D7 BD D7
BD40    BD D7 BD D7 BD D7 BD D7 BD D7 BD D7 BD E3 BD DB
BD50    BD EF BD E7 BD F7 BD DF BD F3 BD E3 BD EB BD 1B
BD60    BE 1F BE 27 BE 27 BE 27 BE 1B BE 23 BE 23 BE 23
BD70    BE FB BD FB BD 0F BE 07 BE 17 BE FF BD 13 BE 03
BD80    BE 0B BE AF BD AF BD B7 BD B3 BD B7 BD AF BD B7
BD90    BD B3 BD B7 BD A7 BD A7 BD AB BD AB BD AB BD A7
BDA0    BD A7 BD A7 BD A7 BD 70 71 72 73 71 70 73 72 D4
BDB0    EF EF EF D0 EF EF EF D2 EF EF EF 00 01 02 03 01
BDC0    00 03 02 14 15 16 17 18 19 1A 1B 0C 0D 0E 0F 0D
BDD0    0C 0F 0E 15 14 17 16 38 39 3A 3B 50 51 52 53 52
BDE0    53 50 51 40 41 42 43 41 40 43 42 48 49 4A 4B 49
BDF0    48 4B 4A 4A 4B 48 49 4B 4A 49 48 A0 A1 A2 A3 A2
BE00    A3 A0 A1 98 99 9A 9B 99 98 9B 9A 9C 9D 9E 9F 9D
BE10    9C 9F 9E 9E 9F 9C 9D 9F 9E 9D 9C 68 69 6A 6B 6C
BE20    6D 6E 6F 60 61 62 63 61 60 63 62 6D 6C 6F 6E 69
BE30    68 6B 6A
```

动作卡通的中断处理系列子程序

```
BE33    8A        TXA
BE34    48        PHA
BE35    98        TYA
BE36    48        PHA
BE37    2044BE    JSR    $BE44
BE3A    68        PLA
BE3B    A8        TAY
BE3C    68        PLA
BE3D    AA        TAX
BE3E    A900      LDA    #$00
BE40    85AB      STA    $AB
BE42    68        PLA
BE43    40        RTI
BE44    A200      LDX    #$00
BE46    86A9      STX    $A9
BE48    A908      LDA    #$08
BE4A    855D      STA    $5D
BE4C    A6A9      LDX    $A9
BE4E    BD4506    LDA    $0645,X
BE51    2910      AND    #$10
BE53    D026      BNE    $BE7B
BE55    C65D      DEC    $5D
BE57    F00A      BEQ    $BE63
BE59    E8        INX
BE5A    E6A9      INC    $A9
BE5C    A5A9      LDA    $A9
```

```
BE5E    C908      CMP    #$08
BE60    D0EA      BNE    $BE4C
BE62    60        RTS
BE63    A900      LDA    #$00
BE65    85AC      STA    $AC
BE67    60        RTS
```

;动作卡通的运动量为零时的处理

```
BE68    BDC506    LDA    $06C5,X
BE6B    9DC306    STA    $06C3,X
BE6E    A6A9      LDX    $A9
BE70    BD4506    LDA    $0645,X
BE73    29EF      AND    #$EF
BE75    9D4506    STA    $0645,X
BE78    4C5ABE    JMP    $BE5A
```

;动作卡通运动控制处理

```
BE7B    A5A9      LDA    $A9
BE7D    0A        ASL
BE7E    0A        ASL
BE7F    0A        ASL
BE80    85A6      STA    $A6
BE82    AA        TAX
BE83    DEC206    DEC    $06C2,X
BE86    D0D2      BNE    $BE5A
BE88    BDC406    LDA    $06C4,X
BE8B    9DC206    STA    $06C2,X
BE8E    DEC306    DEC    $06C3,X
BE91    F0D5      BEQ    $BE68
```

;把卡通定义数据写入二页

```
BE93    BDC106    LDA    $06C1,X
BE96    290F      AND    #$0F
BE98    A8        TAY
BE99    B995BC    LDA    $BC95,Y
BE9C    85DD      STA    $DD
BE9E    98        TYA
BE9F    0A        ASL
BEA0    A8        TAY
BEA1    B975BC    LDA    $BC75,Y
BEA4    85A7      STA    $A7
BEA6    B976BC    LDA    $BC76,Y
```

;取卡通种类

```
BEA9    85A8      STA    $A8
BEAB    BDC006    LDA    $06C0,X
BEAE    0A        ASL
BEAF    A8        TAY
BEB0    B917BD    LDA    $BD17,Y
BEB3    18        CLC
BEB4    7DC606    ADC    $06C6,X
BEB7    9DC606    STA    $06C6,X
BEBA    18        CLC
BEBB    B918BD    LDA    $BD18,Y
BEBE    7DC706    ADC    $06C7,X
```

;取显示座标

```
BEC1    9DC706    STA    $06C7,X
BEC4    A000      LDY    #$00
BEC6    B1A7      LDA    ($A7),Y
BEC8    85D5      STA    $D5
BECA    C8        INY
BECB    B1A7      LDA    ($A7),Y
BECD    85D6      STA    $D6
BECF    BDC006    LDA    $06C0,X
BED2    A8        TAY
BED3    B1D5      LDA    ($D5),Y
BED5    85D7      STA    $D7
BED7    F020      BEQ    $BEF9
BED9    BDC106    LDA    $06C1,X
BEDC    29F0      AND    #$F0
BEDE    18        CLC
BEDF    6910      ADC    #$10
BEE1    C5D7      CMP    $D7
BEE3    D002      BNE    $BEE7
BEE5    A900      LDA    #$00
BEE7    48        PHA
BEE8    A90F      LDA    #$0F
BEEA    3DC106    AND    $06C1,X
BEED    9DC106    STA    $06C1,X
BEF0    68        PLA
BEF1    48        PHA
BEF2    1DC106    ORA    $06C1,X
BEF5    9DC106    STA    $06C1,X
BEF8    68        PLA
BEF9    4A        LSR
BEFA    4A        LSR
BEFB    85D7      STA    $D7
BEFD    A002      LDY    #$02
BEFF    B1A7      LDA    ($A7),Y
BF01    85D9      STA    $D9
BF03    C8        INY
BF04    B1A7      LDA    ($A7),Y
```

```
BF06    85DA      STA   $DA
BF08    BDC006    LDA   $06C0,X
BF0B    0A        ASL
BF0C    A8        TAY
BF0D    B1D9      LDA   ($D9),Y
BF0F    85DB      STA   $DB
BF11    C8        INY
BF12    B1D9      LDA   ($D9),Y
BF14    85DC      STA   $DC
BF16    BDC006    LDA   $06C0,X
BF19    C901      CMP   #$01
BF1B    F004      BEQ   $BF21
BF1D    C905      CMP   #$05
BF1F    D050      BNE   $BF71
;判断卡通种类
BF21    BDC106    LDA   $06C1,X
BF24    290F      AND   #$0F
;是马里欧转$BF4B
BF26    F023      BEQ   $BF4B
BF28    C901      CMP   #$01
;是丽莎转$BF4B
BF2A    F01F      BEQ   $BF4B
BF2C    C904      CMP   #$04
;是马里欧、丽莎、企鹅以外的卡通转$$BF71
BF2E    D041      BNE   $BF71
BF30    A5D7      LDA   $D7
;企鹅的处理
BF32    F033      BEQ   $BF67
BF34    BDC006    LDA   $06C0,X
BF37    C905      CMP   #$05
BF39    F008      BEQ   $BF43
BF3B    A92B      LDA   #$2B
BF3D    85DB      STA   $DB
BF3F    A9BE      LDA   #$BE
BF41    D012      BNE   $BF55
BF43    A92F      LDA   #$2F
BF45    85DB      STA   $DB
BF47    A9BE      LDA   #$BE
BF49    D00A      BNE   $BF55
;马里欧、丽莎的处理
BF4B    A5D7      LDA   $D7
BF4D    F018      BEQ   $BF67
BF4F    A9D3      LDA   #$D3
BF51    85DB      STA   $DB
BF53    A9BD      LDA   #$BD
BF55    85DC      STA   $DC
BF57    A4A9      LDY   $A9
BF59    A940      LDA   #$40
BF5B    194506    ORA   $0645,Y
BF5E    994506    STA   $0645,Y
BF61    A900      LDA   #$00
BF63    85D7      STA   $D7
BF65    F00A      BEQ   $BF71
BF67    A4A9      LDY   $A9
BF69    B94506    LDA   $0645,Y
BF6C    293F      AND   #$3F
BF6E    994506    STA   $0645,Y
;其它卡通的处理
BF71    A5A9      LDA   $A9
BF73    20C5A6    JSR   $A6C5
BF76    AA        TAX
BF77    A000      LDY   #$00
BF79    B1DB      LDA   ($DB),Y
BF7B    18        CLC
BF7C    65DD      ADC   $DD
BF7E    65D7      ADC   $D7
;送第一个卡通块的序号
BF80    9D8102    STA   $0281,X
BF83    C8        INY
BF84    B1DB      LDA   ($DB),Y
BF86    18        CLC
BF87    65DD      ADC   $DD
BF89    65D7      ADC   $D7
;送第二个卡通块的序号
BF8B    9D8502    STA   $0285,X
BF8E    C8        INY
BF8F    B1DB      LDA   ($DB),Y
BF91    18        CLC
BF92    65DD      ADC   $DD
BF94    65D7      ADC   $D7
;送第三个卡通块的序号
BF96    9D8902    STA   $0289,X
BF99    C8        INY
BF9A    B1DB      LDA   ($DB),Y
BF9C    18        CLC
BF9D    65DD      ADC   $DD
BF9F    65D7      ADC   $D7
;送第四个卡通块的序号
```

```
BFA1    9D8D02    STA    $028D,X
BFA4    A4A9      LDY    $A9
BFA6    B94506    LDA    $0645,Y
BFA9    9D8202    STA    $0282,X
BFAC    9D8602    STA    $0286,X
BFAF    9D8A02    STA    $028A,X
;送卡通状态
BFB2    9D8E02    STA    $028E,X
BFB5    A4A6      LDY    $A6
BFB7    B9C606    LDA    $06C6,Y
BFBA    9D8302    STA    $0283,X
BFBD    9D8B02    STA    $028B,X
BFC0    18        CLC
BFC1    6908      ADC    #$08
BFC3    9D8702    STA    $0287,X
;送卡通的X座标
BFC6    9D8F02    STA    $028F,X
BFC9    B9C706    LDA    $06C7,Y
BFCC    9D8002    STA    $0280,X
BFCF    9D8402    STA    $0284,X
BFD2    18        CLC
BFD3    6908      ADC    #$08
BFD5    9D8802    STA    $0288,X
;送卡通的Y座标
BFD8    9D8C02    STA    $028C,X
BFDB    4C5ABE    JMP    $BE5A
BFDE    4C7083    JMP    $8370
BFE1    4C6D83    JMP    $836D
     DEF MOVE 指令的处理
BFE4    20A18C    JSR    $8CA1
BFE7    2089A2    JSR    $A289
BFEA    206D9C    JSR    $9C6D
BFED    203BA7    JSR    $A73B
BFF0    2093A2    JSR    $A293
BFF3    AD0004    LDA    $0400
BFF6    AA        TAX
BFF7    A900      LDA    #$00
BFF9    9D4506    STA    $0645,X
BFFC    8A        TXA
BFFD    85A4      STA    $A4
BFFF    0A        ASL
C000    0A        ASL
C001    0A        ASL
C002    85A5      STA    $A5
C004    20A285    JSR    $85A2
C007    20D48C    JSR    $8CD4
C00A    C98A      CMP    #$8A
C00C    D0D3      BNE    $BFE1
C00E    20A18C    JSR    $8CA1
C011    2089A2    JSR    $A289
;取动作卡通的种类编号
C014    20A2C0    JSR    $C0A2
C017    290F      AND    #$0F
C019    9DC106    STA    $06C1,X
;取动作卡通的运动方向
C01C    209FC0    JSR    $C09F
C01F    C909      CMP    #$09
C021    B0BB      BCS    $BFDE
C023    9DC006    STA    $06C0,X
C026    BDC106    LDA    $06C1,X
C029    0A        ASL
C02A    A8        TAY
C02B    B975BC    LDA    $BC75,Y
C02E    8519      STA    $19
C030    B976BC    LDA    $BC76,Y
C033    851A      STA    $1A
C035    A004      LDY    #$04
C037    B119      LDA    ($19),Y
C039    851B      STA    $1B
C03B    C8        INY
C03C    B119      LDA    ($19),Y
C03E    851C      STA    $1C
C040    BDC006    LDA    $06C0,X
C043    A8        TAY
C044    B11B      LDA    ($1B),Y
C046    A4A4      LDY    $A4
;取动作卡通的状态数据
C048    994506    STA    $0645,Y
C04B    20A8C0    JSR    $C0A8
C04E    B04E      BCS    $C09E
;取动作卡通的运动速度
C050    209FC0    JSR    $C09F
C053    9DC206    STA    $06C2,X
C056    9DC406    STA    $06C4,X
;取动作卡通的运动量
C059    209FC0    JSR    $C09F
C05C    18        CLC
C05D    6901      ADC    #$01
```

```
C05F   9DC306   STA   $06C3,X
C062   9DC506   STA   $06C5,X
C065   20A8C0   JSR   $C0A8
C068   B034     BCS   $C09E
;取动作卡通的显示优先度
C06A   209FC0   JSR   $C09F
C06D   A000     LDY   #$00
C06F   AD0004   LDA   $0400
C072   F002     BEQ   $C076
C074   A020     LDY   #$20
C076   A6A4     LDX   $A4
C078   98       TYA
C079   1D4506   ORA   $0645,X
C07C   9D4506   STA   $0645,X
C07F   20A8C0   JSR   $C0A8
C082   B01A     BCS   $C09E
;取动作卡通的配色号码
C084   209FC0   JSR   $C09F
C087   2903     AND   #$03
C089   A6A4     LDX   $A4
C08B   1D4506   ORA   $0645,X
C08E   9D4506   STA   $0645,X
C091   2093A2   JSR   $A293
C094   A978     LDA   #$78
C096   A6A5     LDX   $A5
C098   9DC606   STA   $06C6,X
C09B   9DC706   STA   $06C7,X
C09E   60       RTS
C09F   2094A5   JSR   $A594
C0A2   206D9C   JSR   $9C6D
C0A5   A6A5     LDX   $A5
C0A7   60       RTS
C0A8   20D48C   JSR   $8CD4
C0AB   C929     CMP   #$29
C0AD   F002     BEQ   $C0B1
C0AF   18       CLC
C0B0   60       RTS
C0B1   20A18C   JSR   $8CA1
C0B4   38       SEC
C0B5   60       RTS
    POSITION 保留字的入口
C0B6   206D9C   JSR   $9C6D
C0B9   203BA7   JSR   $A73B
C0BC   0A       ASL
C0BD   0A       ASL
C0BE   0A       ASL
C0BF   85A5     STA   $A5
;取动作卡通的 X 座标
C0C1   209FC0   JSR   $C09F
C0C4   9DC606   STA   $06C6,X
;取动作卡通的 Y 座标
C0C7   209FC0   JSR   $C09F
C0CA   9DC706   STA   $06C7,X
C0CD   60       RTS
    MOVE 保留字的入口
C0CE   206D9C   JSR   $9C6D
C0D1   203BA7   JSR   $A73B
C0D4   AA       TAX
C0D5   A910     LDA   #$10
C0D7   1D4506   ORA   $0645,X
C0DA   9D4506   STA   $0645,X
C0DD   207883   JSR   $8378
C0E0   F006     BEQ   $C0E8
C0E2   2094A5   JSR   $A594
C0E5   4CCEC0   JMP   $C0CE
C0E8   A9FF     LDA   #$FF
C0EA   85AC     STA   $AC
C0EC   60       RTS
    ERA 保留字的入口
C0ED   A9FF     LDA   #$FF
C0EF   AEA900   LDX   $00A9
    CUT 保留字的入口
C0F0   A900     LDA   #$00
C0F2   8575     STA   $75
C0F4   206D9C   JSR   $9C6D
C0F7   203BA7   JSR   $A73B
C0FA   A8       TAY
C0FB   B94506   LDA   $0645,Y
C0FE   29EF     AND   #$EF
C100   994506   STA   $0645,Y
C103   A575     LDA   $75
C105   F013     BEQ   $C11A
C107   98       TYA
C108   20C5A6   JSR   $A6C5
C10B   A8       TAY
C10C   A9F8     LDA   #$F8
C10E   998002   STA   $0280,Y
C111   998402   STA   $0284,Y
```

```
C114   998802   STA   $0288,Y        C11F   2094A5   JSR   $A594
C117   998C02   STA   $028C,Y        C122   A575     LDA   $75
C11A   207883   JSR   $8378          C124   F0CA     BEQ   $C0F0
C11D   F0CD     BEQ   $C0EC          C126   D0C5     BNE   $C0ED
```

保留字数据区

```
C128                   80 47 4F 54 4F 81 47 4F
                          GOTO        GOSUB
C130  53 55 42 82 52 55 4E 83 52 45 54 55 52 4E 84 52
            RUN      RETURN
C140  45 53 54 4F 52 45 85 54 48 45 4E 86 4C 49 53 54
        RESTORE          THEN        LIST
C150  87 53 59 53 54 45 4D 88 54 4F 89 53 54 45 50 8A
        SYSTEM           TO       STEP
C160  53 50 52 49 54 45 8B 50 52 49 4E 54 8C 46 4F 52
          SPRITE         PRINT        FOR
C170  8D 4E 45 58 54 8E 50 41 55 53 45 8F 49 4E 50 55
          NEXT        PAUSE        INPUT
C180  54 90 4C 49 4E 50 55 54 91 44 41 54 41 92 49 46
             LINPUT          DATA      IF
C190  93 52 45 41 44 94 44 49 4D 95 52 45 4D 96 53 54
          READ      DIM      REM      STOP
C1A0  4F 50 97 43 4F 4E 54 98 43 4C 53 99 43 4C 45 41
              CONT       CLS       CLEAR
C1B0  52 9A 4F 4E 9B 4F 46 46 9C 43 55 54 9D 4E 45 57
          ON       OFF       CUT       NEW
C1C0  9E 50 4F 4B 45 9F 43 47 53 45 54 A0 56 49 45 57
          POKE       CGSET          VIEW
C1D0  A1 4D 4F 56 45 A2 45 4E 44 A3 50 4C 41 59 A4 42
        MOVE         END         PLAY
C1E0  45 45 50 A5 4C 4F 41 44 A6 53 41 56 45 A7 50 4F
      BEEP     LOAD      SAVE
C1F0  53 49 54 49 4F 4E A8 4B 45 59 A9 43 4F 4C 4F 52
      POSITION         KEY      COLOR
C200  AA 44 45 46 AB 43 47 45 4E AC 53 57 41 50 AD 43
        DEF        CGEN        SWAP
C210  41 4C 4C AE 4C 4F 43 41 54 45 AF 50 41 4C 45 54
      CALL     LOCATE          PALET
C220  B0 45 52 41 EF 58 4F 52 F0 4F 52 F1 41 4E 44 F2
        ERA      XOR       OR      AND
C230  4E 4F 54 F3 3C 3E F4 3E 3D F5 3C 3D F6 3D F7 3E
        NOT     <>      >=      <=     =      >
C240  F8 3C F9 2B FA 2D FB 4D 4F 44 FC 2F FD 2A CA 41
      <     +     -      MOD      /      *     ABS
C250  42 53 CB 41 53 43 CC 53 54 52 24 CD 46 52 45 CE
```

```
          ASC          STR $          FRE
C260  4C 45 4E CF 50 45 45 4B D0 52 4E 44 D1 53 47 4E
       LEN        PEEK         RND        SGN
C270  D2 53 50 43 D3 54 41 42 D4 4D 49 44 24 D5 53 54
       SPC         TAB         MID $        STICK
C280  49 43 4B D6 53 54 52 49 47 D7 58 50 4F 53 D8 59
               STRIG              XPOS
C290  50 4F 53 D9 56 41 4C DA 50 4F 53 DB 43 53 52 4C
       YPOS       VAL         POS        CSRLIN
C2A0  49 4E DC 43 48 52 24 DD 48 45 58 24 DE 49 4E 4B
               CHR $          HEX $       INKEY $
C2B0  45 59 24 DF 52 49 47 48 54 24 E0 4C 45 46 54 24
                 RIGHT $             LEFT $
C2C0  E1 53 43 52 24 FF 46 91 C6 90 7C 85 26 91 E0 98
        SCR $
C2D0  6D 83 AF 85 D7 80 6D 83 6D 83 F3 A7 40 87 7E 92
C2E0  0D 93 98 91 C2 93 F6 94 43 83 04 92 4D 99 4E 96
C2F0  43 83 DA 86 17 87 75 AB 26 92 4D 95 6D 83 F0 C0
C300  82 83 DF 91 46 A7 32 A9 CE C0 0B 82 1D 9A 0F B3
C310  FC 97 8C 97 B6 C0 AA 95 4E BA 87 A8 CA A6 96 94
C320  8C 91 CA 91 EA A6 ED C0 69 A3 0A A4 85 A4 1A A3
C330  FF A3 58 A3 9F A3 79 A3 6D 83 6D 83 FB A5 AA A6
C340  B9 A6 46 A3 4F A3 BE A4 2E A3 F3 A2 61 A4 B5 A4
C350  20 A5 BD A5 74 A5 1D A4 46 20 42 41 53 49 43 20
                                F  B A S I C
C360  42 53 2E 32 41 0D 0A B4 20 43 4E 45 44 43 20 53
      B S . 2 A               C N E D C    S
C370  48 45 4E 20 5A 48 45 4E 20 43 4F 2E 20 20 0D 0A
      H E N   Z H E N    C O .
C380  00 00 00 00 20 20 20 46 20 42 41 53 49 43 20 20
                             F  B A S I C
C390  00 08 03 84 C3 07 07 6E 81 07 09 77 81 07 0B 85
        BASIC   1—BASIC   2—BGGRAPHIC 3—END
C3A0  81 06 12 8C 81 AF C3 BF C3 CF C3 DF C3 EF C3 0F
        1,2,3 KEY IN !!
C3B0  2C 15 07 0F 27 21 12 0F 29 36 17 0F 30 26 07 0F
C3C0  30 21 02 0F 30 27 18 0F 30 27 16 0F 29 36 17 0F
C3D0  36 16 02 0F 27 30 19 0F 35 25 17 0F 30 27 16 0F
C3E0  30 16 01 0F 10 00 01 0F 30 29 09 0F 30 16 07 0F
C3F0  30 26 12 0F 30 15 12 0F 30 12 16 0F 30 26 19 FF
```

系统复位入口

```
C400  78      SEI
C401  D8      CLD
C402  A910    LDA   #$10
C404  8D0020  STA   $2000
C407  A2FF    LDX   #$FF
C409  9A      TXS
```

```
C40A  AD0220   LDA  $2002
C40D  2980     AND  #$80
C40F  EA       NOP
C410  EA       NOP
C411  A007     LDY  #$07
C413  8401     STY  $01
C415  A000     LDY  #$00
C417  8400     STY  $00
C419  98       TYA
C41A  9100     STA  ($00),Y
C41C  88       DEY
C41D  D0FB     BNE  $C41A
C41F  C601     DEC  $01
C421  10F7     BPL  $C41A
C423  A9FF     LDA  #$FF
C425  857A     STA  $7A
C427  2094CD   JSR  $CD94
C42A  A9FD     LDA  #$FD
C42C  85E0     STA  $E0
C42E  A95F     LDA  #$5F
C430  85E1     STA  $E1
C432  A510     LDA  $10
C434  4980     EOR  #$80
C436  207AEC   JSR  $EC7A
C439  A90F     LDA  #$0F
C43B  8D1540   STA  $4015
C43E  A94C     LDA  #$4C
C440  85ED     STA  $ED
C442  A9A7     LDA  #$A7
C444  85EE     STA  $EE
C446  A9CD     LDA  #$CD
C448  85EF     STA  $EF
C44A  A560     LDA  $60
C44C  C901     CMP  #$01
C44E  D00A     BNE  $C45A
C450  A510     LDA  $10
C452  297F     AND  #$7F
C454  207AEC   JSR  $EC7A
C457  4C0080   JMP  $8000
C45A  20E8ED   JSR  $EDE8
C45D  A541     LDA  $41
C45F  F014     BEQ  $C475
C461  20A6DF   JSR  $DFA6
C464  A5B9     LDA  $B9
C466  F006     BEQ  $C46E
C468  2001DE   JSR  $DE01
C46B  4C71C4   JMP  $C471
C46E  20A9DD   JSR  $DDA9
C471  A900     LDA  #$00
C473  8541     STA  $41
C475  4C4AC4   JMP  $C44A
```

复位画面绘制的数据区

```
C478                                   C5 F0 00 38 C5 F1 00 68
C480  C5 F2 00 A0 C5 F3 00 D0 0F 07 73 AB F8 02 DE B0
C490  B6 BC C2 10 DF E8 38 70 A8 BF 63 73 83 90 62 72
C4A0  82 92 0C 0C 0C 1C 16 06 07 08 0F 08 07 06 00 70
C4B0  10 20 20 21 21 21 21 21 21 22 22 22 C6 E6 06 26
C4C0  46 66 86 A6 46 66 86 E1 E1 E1 E1 E1 E5 E9 EE F3
C4D0  F7 FB FF 03 C4 C4 C4 C4 C4 C4 C4 C4 C4 C4 C4 C4
C4E0  C5 0C 3F 07 C5 0C 3F 0C C5 68 24 F4 61 00 68 84
C4F0  FA 61 18 C6 20 50 03 14 3F 50 03 44 23 11 C5 67
C500  22 11 C5 44 23 17 C5 14 0F 16 02 2A 14 0F 0F 0F
C510  0F 15 20 20 20 20 20 15 FF FF FF FF FF C6 CA CE
C520  D2 D6 CC D0 D4 FF FE 20 FD
```

对话工作数据区

```
C529                                      A1 B5 C2 C9 CD D4 E2
C530  FC 07 13 22 29 47 55 24 80 86 94 24 24 45 CA DB
C540  24 24 F8 24 24 24 24 24 24 24 24 24 24 24 24 27
C550  24 24 24 24 24 24 AB 24 7C 8A 24 24 B4 B7 BE CF
```

C560	D2	E1	9D	24	24	C7	C5	C5	C5	C5	C5	C5	C5	C6	C6	C6
C570	C6	C6	C6	C7	C6	C6	C6	C7	C7	C9	C6	C6	C7	C7	C6	C7
C580	C7	C7	C7	C7	C7	C7	C7	C7	C7	C7	C7	C7	C7	C7	C7	C7
C590	C7	C7	C6	C7	C9	C9	C7	C7	C9	C9	C9	C9	C9	C9	C9	C7
C5A0	C7	49	27	4D	0D	20	46	43	53	39	30	20	43	4F	4D	50
		I	'	M			F	C	S	9	0		C	O	M	P
C5B0	55	54	45	52	FF	4E	4F	57	20	57	45	20	53	54	41	52
	U	T	E	R		N	O	W		W	E		S	T	A	R
C5C0	54	FF	4F	4B	20	20	20	20	FF	59	45	53	FF	20	4F	4B
	T		O	K						Y	E	S			O	K
C5D0	21	FF	21	FF	4E	4F	57	20	57	45	20	53	54	41	52	54
	!		!		N	O	W		W	E		S	T	A	R	T
C5E0	21	FF	57	48	4F	20	41	52	45	20	59	4F	55	3F	0D	49
	!		W	H	O		A	R	E		Y	O	U	?		I
C5F0	4E	50	55	54	20	4E	41	4D	45	FF	61	FF	4B	45	59	20
	N	P	U	T		N	A	M	E				K	E	Y	
C600	41	47	41	49	4E	21	FF	22	C2	22	20	0D	20	4E	45	45
	A	G	A	I	N	!								N	E	E
C610	44	3F	FF	4E	4F	57	20	57	45	2E	2E	2E	20	20	20	20
	D	?		N	O	W		W	E	.	.	.				
C620	20	FF	42	59	45	42	59	45	FF	43	41	53	53	45	54	54
			B	Y	E	B	Y	E		C	A	S	S	E	T	T
C630	45	27	53	20	53	57	49	54	43	48	0D	4F	46	46	20	50
	E	'	S		S	W	I	T	C	H		O	F	F		P
C640	4C	45	41	53	45	21	FF	22	C2	22	20	0D	20	53	54	41
	L	E	A	S	E	!								S	T	A
C650	52	54	FF	6C	FF	22	C2	22	20	0D	20	4E	4F	57	20	46
	R	T										N	O	W		F
C660	49	4E	49	53	48	45	44	FF	6F	FF	60	60	21	0D	7E	63
	I	N	I	S	H	E	D									
C670	75	20	64	60	61	6B	6F	69	73	97	20	60	87	7E	6C	FF
C680	48	45	4C	4C	4F	FF	4E	45	58	54	20	53	45	4C	45	43
	H	E	L	L	O		N	E	X	T		S	E	L	E	C
C690	54	2E	FF	FF	4F	4B	21	20	53	4F	52	52	59	21	FF	3A
	T				O	K	!		S	O	R	R	Y	!		
C6A0	7B	8E	8C	7B	8E	2D	89	3A	8D	20	FF	4E	4F	57	20	57
												N	O	W		W
C6B0	49	4C	4C	20	42	45	20	20	20	20	20	20	20	20	43	4F
	I	L	L		B	E									C	O
C6C0	4E	54	49	4E	55	45	20	21	20	FF	C0	0D	20	48	41	56
	N	T	I	N	U	E		!						H	A	V
C6D0	45	20	41	20	52	45	53	54	20	3F	FF	54	49	4D	45	20
	E		A		R	E	S	T		?		T	I	M	E	
C6E0	4F	56	45	52	FF	82	61	A1	6B	7E	6B	6F	21	FF	6E	89

	O	V	E	R												
C6F0	A4	79	20	60	72	7E	6C	FF	48	41	56	45	20	41	20	52
										H	A	V	E	A		R
C700	45	53	54	2E	0D	20	20	20	20	20	4F	52	20	20	20	20
	E	S	T								O	R				
C710	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
C720	20	20	20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
C730	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
C740	FF	62	86	74	61	8D	20	79	9D	81	7E	6C	FF	7E	A1	20
C750	62	7E	89	72	20	61	7E	6D	8C	FF	6C	A4	75	20	64	74
C760	67	74	87	75	20	74	86	89	72	0D	61	88	A4	6B	95	62
C770	21	FF	65	86	A1	78	20	70	95	62	6B	97	20	85	61	78
C780	A4	0D	6C	AF	2D	71	20	6B	9B	73	20	A9	8C	66	95	62
C790	0D	6C	A9	72	75	20	97	8C	A6	87	97	20	66	66	7E	6C
C7A0	FF	49	27	4D	20	43	4E	45	44	43	20	46	41	4D	49	4C
C7B0	59	20	20	43	4F	4D	50	55	54	45	52	20	53	59	53	54
C7C0	45	4D	FF	66	8D	71	68	72	20	67	A1	6A	61	FF	6F	61
C7D0	87	95	67	97	20	6C	69	6B	20	85	8B	96	72	61	7E	6C
C7E0	80	87	8D	20	6B	74	61	85	62	75	20	69	69	8A	97	68
C7F0	72	0D	67	A1	6A	61	FF	66	A8	8C	79	20	6E	62	65	61
C800	0D	6D	96	66	95	67	72	66	75	0D	74	75	9B	73	82	20
C810	6C	6C	81	72	67	A1	6A	61	FF	6D	61	6B	8C	72	66	75
C820	20	7B	60	8C	72	61	A4	6C	0D	61	86	61	86	20	6B	74
C830	61	85	62	75	20	66	8D	71	68	72	0D	67	A1	6A	61	FF
C840	66	A8	8C	97	20	6A	63	7E	6D	8C	0D	6F	61	9D	8C	20
C850	65	8C	68	61	75	0D	66	8D	20	71	68	72	20	67	A1	6A
C860	61	FF	47	4F	4F	44	21	20	59	4F	55	20	41	52	45	20
			G	O	O	D	!		Y	O	U		A	R	E	
C870	41	20	44	49	4C	49	47	45	4E	54	20	43	48	49	4C	44
	A		D	I	L	I	G	E	N	T		C	H	I	L	D
C880	20	FF	20	65	61	68	71	20	6B	7E	6C	FF	64	64	66	74
C890	20	7F	6C	8D	20	64	65	6B	83	6C	61	0D	7A	A4	6C	0D
C8A0	62	96	65	87	20	6B	74	61	85	62	75	20	66	8D	20	71
C8B0	68	72	67	A1	6A	61	FF	60	6F	7E	78	20	65	61	72	8C
C8C0	97	20	75	A8	96	72	61	7E	6C	78	62	87	71	97	20	60
C8D0	97	87	7E	6D	8C	65	86	0D	80	87	75	20	60	6F	7E	8D
C8E0	20	71	65	8B	74	61	20	85	62	75	FF	6E	78	7A	78	20
C8F0	62	86	74	61	79	20	71	98	78	20	85	62	75	74	87	7E
C900	6B	6F	FF	62	86	74	61	20	64	8B	87	FF	74	65	74	65
C910	20	85	61	20	68	96	65	97	20	A4	7E	6B	6F	0D	85	65
C920	96	6F	20	A4	6C	77	21	FF	60	7E	87	20	85	67	60	87
C930	7E	6D	8C	77	0D	66	8D	20	64	73	6A	74	61	A4	20	67
C940	A1	6A	61	21	FF	43	4E	45	44	43	20	53	48	45	4E	5A
C950	48	45	4E	20	43	4F	2E	54	45	4C	3A	33	36	31	31	30

C960	35	20	33	36	35	34	30	32	20	43	41	42	4C	45	3A	34
C970	37	30	33	20	20	20	20	20	20	20	20	FF	4F	4B	21	0D
													O	K	!	
C980	C0	0D	49	53	20	59	4F	55	2E	FF	C0	0D	44	4F	20	59
			I	S		Y	O	U					D	O		Y
C990	4F	55	52	20	48	4F	4D	45	57	4F	52	4B	FF	50	52	45
	O	U	R		H	O	M	E	W	O	R	K		P	R	E
C9A0	53	53	49	4E	47	20	52	45	53	45	54	20	20	20	20	43
	S	S	I	N	G		R	E	S	E	T					C
C9B0	4F	4D	50	55	54	45	52	20	57	49	4C	4C	20	57	4F	52
	O	M	P	U	T	E	R		W	I	L	L		W	O	R
C9C0	4B	20	41	47	41	49	4E	FF	75	0D	74	87	7E	6C	FF	C0
	K		A	G	A	I	N									
C9D0	21	FF	0D	C0	0D	48	4F	57	20	41	52	45	20	59	4F	55
						H	O	W		A	R	E		Y	O	U
C9E0	FF	43	4E	45	44	43	20	53	48	45	4E	5A	48	45	4E	20
		C	N	E	D	C		S	H	E	N	Z	H	E	N	
C9F0	43	4F	2E	20	49	53	20	47	4C	41	44	20	54	4F	20	53
	C	O	.		I	S		G	L	A	D		T	O		S
CA00	45	52	56	45	20	46	4F	52	20	59	4F	55	2E	20	20	20
	E	R	V	E		F	O	R		Y	O	U	.			
CA10	20	20	20	20	20	20	20	20	20	20	20	20	54	48	41	4E
													T	H	A	N
CA20	4B	53	20	21	FF	6A	61	FF	62	86	74	61	8D	20	71	A3
	K	S														
CA30	68	7E	6C	65	3F	FF	01	06	00	FF	22	04	00	FF	00	08
CA40	C0	FF	01	08	C0	FF	0A	12	0A	0A	10	40	20	40	40	20
CA50	5C	60	5C	58	CA	CA	CA	CA	04	05	0D	03	10	10	10	10
CA60	04	05	10	10	44	4A	51	57	67	01	F4	A5	7D	FA	22	32
CA70	42	77	CC	03	D7	D9	CA	D9	DA	DA	DA	DA	DA	23	DB	42
CA80	00	21	CC	06	52	41	4D	20	4F	4B	00	55	AA	00	13	26
CA90	B7	BB	BE	C2	CD	D9	E4	EE	F2	F8	FD	03	09	09	0A	10
CAA0	15	1A	70	70	70	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA
CAB0	CB	CB	CB	CB	CB	CB	CB	45	4E	44	0D	4F	4B	0D	4E	4F
								E	N	D		O	K		N	O
CAC0	0D	0D	46	20	42	41	53	49	43	0D	49	43	0D	43	41	4C
			F	B		A	S	I	C		I	C		C	A	L
CAD0	43	55	4C	41	54	4F	52	0D	0D	4D	55	53	49	43	42	4F
	C	U	L	A	T	O	R			M	U	S	I	C	B	O
CAE0	41	52	44	0D	4D	45	53	53	42	4F	41	52	44	0D	45	4E
	A	R	D		M	E	S	S	B	O	A	R	D		E	N
CAF0	44	0D	48	45	4C	4C	4F	0D	48	49	21	0D	0D	48	45	4C
	D		H	E	L	L	O		H	I	!			H	E	L
CB00	4C	4F	0D	48	41	48	41	21	0D	0D	42	41	53	49	43	0D

	L	O		H	A	H	A	!			B	A	S	I	C	
CB10	43	41	4C	2E	0D	4D	55	53	2E	0D	4D	45	53	2E	0D	65
	C	A	L	.		M	U	S	.		M	E	S	.		
CB20	67	6B	94	20	AA	2D	A5	0D	88	8E	92	BA	15	C6	CB	CB
CB30	E6	E6	EB	00	07	0B	22	98	2E	3A	3A	3F	3F	48	48	48
CB40	4B	92	92	D8	9B	9F	E4	A7	AF	B4	B9	B9	BE	BE	BE	BE
CB50	DA	EC	E6	DD	BE	FD	FD	A1	CB	CB	CB	CB	CC	CB	CB	CB
CB60	CB	CB	CB	CC	CC	CC	CC	CB	CC	CC	CC	CC	CC	CC	CC	CC
CB70	CC	CC	CC	CB	CC	CC	CB	CC	CC	CC	CC	CC	CC	CC	CC	CC
CB80	CC	CC	CC	CC	CC	CC	CC	CB	09	00	08	01	0F	00	1E	00
CB90	0A	01	06	00	00	06	00	01	14	00	06	00	0B	05	06	FF
CBA0	0F	06	00	06	05	03	10	03	13	00	0D	00	06	00	04	06
CBB0	00	2F	11	00	06	00	0E	06	00	0F	06	00	05	04	00	06
CBC0	00	1A	17	00	FF	06	06	00	10	20	00	06	00	08	05	03
CBD0	0C	03	15	00	0C	00	12	00	0C	00	06	00	04	06	00	0C
CBE0	05	07	09	00	12	00	0C	00	06	00	08	0C	00	06	00	09
CBF0	06	00	19	1F	00	06	00	30	06	00	0A	06	00	2E	FF	01
CC00	0C	00	06	00	02	FF	0A	0E	00	01	00	1C	00	0E	02	08
CC10	05	0F	00	02	00	09	00	0A	03	08	01	0F	00	06	00	0D
CC20	FF	05	08	04	08	07	1D	00	0F	00	0B	00	FF	04	08	09
CC30	08	06	25	00	0F	00	26	00	FF	04	0C	00	06	00	2C	06
CC40	00	39	05	03	19	14	18	00	06	00	04	0C	00	06	00	3A
CC50	05	03	19	18	06	00	04	06	00	1B	05	05	21	10	06	00
CC60	02	06	00	18	06	84	31	03	9A	32	06	50	33	03	BC	34
CC70	06	00	13	22	00	06	D0	35	06	00	27	06	00	12	05	04
CC80	1B	10	23	00	23	00	23	00	23	00	06	00	28	24	00	06
CC90	00	2A	06	00	3B	05	03	0C	01	1A	1A	0C	00	FF	2A	06
CCA0	00	11	06	00	07	12	00	06	00	1C	06	00	07	FF	13	06
CCB0	00	1D	FF	1A	06	00	29	FF	19	06	00	04	12	00	0C	00
CCC0	06	00	03	06	70	36	1F	00	03	00	37	06	00	00	1F	00
CCD0	06	00	38	1F	00	FF	28	00	FF	29	06	00	14	06	00	15
CCE0	05	03	0C	01	1A	2B	06	00	04	06	00	2B	06	00	13	06
CCF0	00	2D	06	00	13	06	00	17	06	00	05	27	00	0C	00	06
CD00	00	16	FF	0A	0B	00	00	00	00	00	00	0B	2C	2C	2C	2C
CD10	00	00	00	00	00	00	0C	0D	10	0E	00	3D	DC	DE	20	30
CD20	CD	50	E0	A4	A7	D4	D7	AA	1F	C2	CD	D9	E4	0A	05	06
CD30	00	00	05	05	00	05	73	00	00	00	00	00	CB	CA	CA	CA
CD40	CA	CB	00	12	24	93	CD	77	DC	45	CD	5B	CF	D5	CE	A4
CD50	D2	78	CF	F8	CE	06	CF	15	D5	97	DB	4A	D4	1F	CF	F8
CD60	CE	18	CF	E8	D0	2A	D0	07	D1	C4	CF	2D	CF	6E	D0	10
CD70	CF	DA	D0	34	F6	0D	FA	46	F6	7A	F6	0F	D1	24	D1	B1
CD80	F4	E8	CE	0C	D1	89	F6	8E	F9	AF	F9	E2	F9	39	D1	23
CD90	E4	03	D1													

复位子程序

```
CD93   60        RTS
CD94   A900      LDA   #$00
CD96   8D1140    STA   $4011
CD99   A906      LDA   #$06
CD9B   8D0120    STA   $2001
CD9E   205AEC    JSR   $EC5A
CDA1   206FEC    JSR   $EC6F
CDA4   4C69EC    JMP   $EC69
       对话工作系统的中断程序入口
CDA7   08        PHP
CDA8   48        PHA
CDA9   8A        TXA
CDAA   48        PHA
CDAB   98        TYA
CDAC   48        PHA
CDAD   A20F      LDX   #$0F
CDAF   B500      LDA   $00,X
CDB1   48        PHA
CDB2   CA        DEX
CDB3   10FA      BPL   $CDAF
CDB5   A510      LDA   $10
CDB7   297F      AND   #$7F
CDB9   207AEC    JSR   $EC7A
CDBC   A578      LDA   $78
CDBE   D009      BNE   $CDC9
CDC0   A511      LDA   $11
CDC2   091E      ORA   #$1E
CDC4   8D0120    STA   $2001
CDC7   8511      STA   $11
CDC9   A900      LDA   #$00
CDCB   8D0320    STA   $2003
CDCE   A902      LDA   #$02
CDD0   8D1440    STA   $4014
CDD3   A68A      LDX   $8A
CDD5   BD69CA    LDA   $CA69,X
CDD8   8500      STA   $00
CDDA   BD73CA    LDA   $CA73,X
CDDD   8501      STA   $01
;绘制各版画面及送显各种内容
CDDF   2043EE    JSR   $EE43
CDE2   A900      LDA   #$00
CDE4   858A      STA   $8A
CDE6   8D0003    STA   $0300
CDE9   8D0103    STA   $0301
CDEC   A57B      LDA   $7B
CDEE   F003      BEQ   $CDF3
;进入 F BASIC 系统
CDF0   4C0080    JMP   $8000
CDF3   A546      LDA   $46
CDF5   F008      BEQ   $CDFF
CDF7   2048CF    JSR   $CF48
CDFA   F003      BEQ   $CDFF
;返回复位程序入口
CDFC   4C00C4    JMP   $C400
CDFF   E63B      INC   $3B
CE01   A551      LDA   $51
CE03   D004      BNE   $CE09
CE05   A90F      LDA   #$0F
CE07   8551      STA   $51
;读键盘
CE09   200380    JSR   $8003
CE0C   8537      STA   $37
CE0E   A67A      LDX   $7A
CE10   F00A      BEQ   $CE1C
CE12   C954      CMP   #$54
CE14   D006      BNE   $CE1C
CE16   2046CE    JSR   $CE46
CE19   4C29CE    JMP   $CE29
CE1C   C903      CMP   #$03
CE1E   F009      BEQ   $CE29
CE20   20D2ED    JSR   $EDD2
;发声处理
CE23   2006FC    JSR   $FC06
;对话工作系统处理
CE26   206ECE    JSR   $CE6E
CE29   AD0220    LDA   $2002
CE2C   A510      LDA   $10
CE2E   4980      EOR   #$80
CE30   207AEC    JSR   $EC7A
;中断现场恢复处理
CE33   A200      LDX   #$00
CE35   867A      STX   $7A
CE37   68        PLA
CE38   9500      STA   $00,X
CE3A   E8        INX
CE3B   E010      CPX   #$10
CE3D   D0F8      BNE   $CE37
CE3F   68        PLA
```

```
CE40    A8        TAY
CE41    68        PLA
CE42    AA        TAX
CE43    68        PLA
CE44    28        PLP
CE45    40        RTI
CE46    A201      LDX    #$01
CE48    A070      LDY    #$70
CE4A    8401      STY    $01
CE4C    A000      LDY    #$00
CE4E    8400      STY    $00
CE50    BD8BCA    LDA    $CA8B,X
CE53    9100      STA    ($00),Y
CE55    D100      CMP    ($00),Y
CE57    D012      BNE    $CE6B
CE59    88        DEY
CE5A    D0F4      BNE    $CE50
CE5C    E601      INC    $01
CE5E    A978      LDA    #$78
CE60    C501      CMP    $01
CE62    D0EC      BNE    $CE50
CE64    CA        DEX
CE65    10E1      BPL    $CE48
CE67    A903      LDA    #$03
CE69    858A      STA    $8A
CE6B    857B      STA    $7B
CE6D    60        RTS
```

对话工作系统主控程序

```
CE6E    2082CE    JSR    $CE82
CE71    A560      LDA    $60
CE73    0A        ASL
CE74    AA        TAX
CE75    BD43CD    LDA    $CD43,X
CE78    8502      STA    $02
CE7A    BD44CD    LDA    $CD44,X
CE7D    8503      STA    $03
CE7F    6C0200    JMP    ($0002)
```

$60、$6B、$64赋值子程序

```
CE82    A560      LDA    $60
CE84    F005      BEQ    $CE8B
CE86    A900      LDA    #$00
CE88    8573      STA    $73
CE8A    60        RTS
CE8B    A900      LDA    #$00
CE8D    85C6      STA    $C6
CE8F    A901      LDA    #$01
CE91    8573      STA    $73
CE93    A662      LDX    $62
CE95    BD28CB    LDA    $CB28,X
CE98    8506      STA    $06
CE9A    BD58CB    LDA    $CB58,X
CE9D    8507      STA    $07
CE9F    A464      LDY    $64
CEA1    B106      LDA    ($06),Y
CEA3    C903      CMP    #$03
CEA5    D006      BNE    $CEAD
CEA7    A2FF      LDX    #$FF
CEA9    86C6      STX    $C6
CEAB    A906      LDA    #$06
CEAD    C9FF      CMP    #$FF
CEAF    F018      BEQ    $CEC9
CEB1    8560      STA    $60
CEB3    AA        TAX
CEB4    C8        INY
CEB5    B106      LDA    ($06),Y
CEB7    856B      STA    $6B
CEB9    E006      CPX    #$06
CEBB    D007      BNE    $CEC4
CEBD    C8        INY
CEBE    B106      LDA    ($06),Y
CEC0    85C3      STA    $C3
CEC2    E664      INC    $64
CEC4    E664      INC    $64
CEC6    E664      INC    $64
CEC8    60        RTS
CEC9    C8        INY
CECA    B106      LDA    ($06),Y
CECC    8562      STA    $62
CECE    A900      LDA    #$00
CED0    8564      STA    $64
CED2    4C8BCE    JMP    $CE8B
```

延时子程序处理

```
CED5    A573      LDA    $73
CED7    F004      BEQ    $CEDD
CED9    A56B      LDA    $6B
CEDB    852D      STA    $2D
CEDD    2009D7    JSR    $D709
CEE0    A52D      LDA    $2D
```

```
CEE2    D003      BNE    $CEE7
CEE4    2013CF    JSR    $CF13
CEE7    60        RTS
CEE8    A912      LDA    #$12
CEEA    8582      STA    $82
CEEC    A577      LDA    $77
CEEE    C9FF      CMP    #$FF
CEF0    D003      BNE    $CEF5
CEF2    20C2CE    JSR    $CEC2
CEF5    4C13CF    JMP    $CF13
CEF8    A901      LDA    #$01
CEFA    8578      STA    $78
CEFC    2080EC    JSR    $EC80
CEFF    A66B      LDX    $6B
CF01    868A      STX    $8A
CF03    4C13CF    JMP    $CF13
CF06    A901      LDA    #$01
CF08    8578      STA    $78
;关显示
CF0A    2080EC    JSR    $EC80
;清屏
CF0D    202DED    JSR    $ED2D
;清卡通定义区
CF10    205AEC    JSR    $EC5A
CF13    A900      LDA    #$00
CF15    8560      STA    $60
CF17    60        RTS
CF18    A900      LDA    #$00
CF1A    8578      STA    $78
CF1C    8560      STA    $60
CF1E    60        RTS
CF1F    A212      LDX    #$12
CF21    BD3D06    LDA    $063D,X
CF24    9D0005    STA    $0500,X
CF27    CA        DEX
CF28    10F7      BPL    $CF21
CF2A    4C13CF    JMP    $CF13
CF2D    2048CF    JSR    $CF48
CF30    D013      BNE    $CF45
CF32    8546      STA    $46
CF34    A564      LDA    $64
CF36    18        CLC
CF37    6907      ADC    #$07
CF39    8564      STA    $64
CF3B    AD3A70    LDA    $703A
CF3E    C95A      CMP    #$5A
CF40    F003      BEQ    $CF45
CF42    2060D2    JSR    $D260
CF45    4C13CF    JMP    $CF13
CF48    A955      LDA    #$55
CF4A    8D3970    STA    $7039
CF4D    CD3970    CMP    $7039
CF50    D008      BNE    $CF5A
CF52    A9AA      LDA    #$AA
CF54    8D3970    STA    $7039
CF57    CD3970    CMP    $7039
CF5A    60        RTS
CF5B    AD3A70    LDA    $703A
CF5E    C95A      CMP    #$5A
CF60    D00C      BNE    $CF6E
CF62    AD3B70    LDA    $703B
CF65    C9CC      CMP    #$CC
CF67    D008      BNE    $CF71
CF69    A90E      LDA    #$0E
CF6B    4C73CF    JMP    $CF73
CF6E    20C2CE    JSR    $CEC2
CF71    A913      LDA    #$13
CF73    8588      STA    $88
CF75    4C13CF    JMP    $CF13
CF78    A573      LDA    $73
CF7A    F003      BEQ    $CF7F
;令$63D~$7D3=H20
CF7C    20FBEE    JSR    $EEFB
CF7F    A537      LDA    $37
CF81    C90D      CMP    #$0D
CF83    F020      BEQ    $CFA5
CF85    A968      LDA    #$68
CF87    85FC      STA    $FC
CF89    A203      LDX    #$03
;键值的判断处理
CF8B    202CEF    JSR    $EF2C
;控制 OPERATOR 栏的光标
CF8E    2046F5    JSR    $F546
;改变卡通的显示状态
CF91    207BD7    JSR    $D77B
;改变卡通和 PPU 配色
CF94    2009D7    JSR    $D709
CF97    A537      LDA    $37
```

```
CF99    F005      BEQ    $CFA0
CF9B    A910      LDA    #$10
CF9D    8524      STA    $24
CF9F    60        RTS
CFA0    A900      LDA    #$00
CFA2    4C75D7    JMP    $D775
CFA5    A910      LDA    #$10
CFA7    85FE      STA    $FE
CFA9    A900      LDA    #$00
CFAB    85FC      STA    $FC
CFAD    A2FF      LDX    #$FF
CFAF    BD3D06    LDA    $063D,X
CFB2    C920      CMP    #$20
CFB4    D003      BNE    $CFB9
CFB6    CA        DEX
CFB7    D0F6      BNE    $CFAF
CFB9    E8        INX
CFBA    862F      STX    $2F
CFBC    A90D      LDA    #$0D
CFBE    9D3D06    STA    $063D,X
CFC1    4C13CF    JMP    $CF13
```

系统名词翻译子程序

```
CFC4    A214      LDX    #$14
CFC6    8600      STX    $00
CFC8    A600      LDX    $00
CFCA    20E6CF    JSR    $CFE6
CFCD    C900      CMP    #$00
CFCF    F00B      BEQ    $CFDC
CFD1    C600      DEC    $00
CFD3    A902      LDA    #$02
CFD5    C500      CMP    $00
CFD7    D0EF      BNE    $CFC8
CFD9    4C13CF    JMP    $CF13
CFDC    A600      LDX    $00
CFDE    BD01CD    LDA    $CD01,X
CFE1    F0F6      BEQ    $CFD9
CFE3    4C9CD2    JMP    $D29C
CFE6    BD8DCA    LDA    $CA8D,X
CFE9    8502      STA    $02
CFEB    BDA2CA    LDA    $CAA2,X
CFEE    8503      STA    $03
CFF0    A2FF      LDX    #$FF
CFF2    A0FF      LDY    #$FF
CFF4    86B4      STX    $B4
CFF6    E8        INX
CFF7    BD3D06    LDA    $063D,X
CFFA    C90D      CMP    #$0D
CFFC    F026      BEQ    $D024
CFFE    C920      CMP    #$20
D000    F0F4      BEQ    $CFF6
D002    C8        INY
D003    B102      LDA    ($02),Y
D005    C920      CMP    #$20
D007    F0F9      BEQ    $D002
D009    BD3D06    LDA    $063D,X
D00C    D102      CMP    ($02),Y
D00E    D014      BNE    $D024
D010    E6B4      INC    $B4
D012    86B5      STX    $B5
D014    A6B4      LDX    $B4
D016    9D5005    STA    $0550,X
D019    A6B5      LDX    $B5
D01B    C90D      CMP    #$0D
D01D    F008      BEQ    $D027
D01F    E8        INX
D020    C8        INY
D021    4C09D0    JMP    $D009
D024    A9FF      LDA    #$FF
D026    60        RTS
D027    A900      LDA    #$00
D029    60        RTS
```

对话板处理程序

```
D02A    A902      LDA    #$02
D02C    8500      STA    $00
D02E    8577      STA    $77
D030    A600      LDX    $00
D032    20E6CF    JSR    $CFE6
D035    C900      CMP    #$00
D037    F021      BEQ    $D05A
D039    C600      DEC    $00
D03B    10F3      BPL    $D030
D03D    A200      LDX    #$00
D03F    BD1370    LDA    $7013,X
D042    9D0070    STA    $7000,X
D045    E8        INX
D046    E026      CPX    #$26
D048    D0F5      BNE    $D03F
D04A    A212      LDX    #$12
```

```
D04C   BD3D06   LDA   $063D,X
D04F   9D2670   STA   $7026,X
D052   CA       DEX
D053   10F7     BPL   $D04C
D055   A903     LDA   #$03
D057   4C9CD2   JMP   $D29C
D05A   4C13CF   JMP   $CF13
       各板询问时,操作者输入内容的
       判断检索
D05D   A573     LDA   $73
D05F   F004     BEQ   $D065
D061   A9FF     LDA   #$FF
D063   852D     STA   $2D
D065   20BAD4   JSR   $D4BA
D068   2078CF   JSR   $CF78
D06B   A560     LDA   $60
D06D   60       RTS
       根据操作者的输入内容转移到相应的
       处理程序入口
D06E   A907     LDA   #$07
D070   8587     STA   $87
D072   8545     STA   $45
D074   205DD0   JSR   $D05D
D077   F003     BEQ   $D07C
D079   4CF7D0   JMP   $D0F7
;系统名词的翻译
D07C   20C4CF   JSR   $CFC4
D07F   A564     LDA   $64
D081   F039     BEQ   $D0BC
D083   A500     LDA   $00
D085   C904     CMP   #$04
D087   F024     BEQ   $D0AD
D089   C905     CMP   #$05
D08B   F030     BEQ   $D0BD
D08D   C906     CMP   #$06
D08F   3008     BMI   $D099
D091   C90A     CMP   #$0A
D093   3009     BMI   $D09E
D095   C911     CMP   #$11
D097   1008     BPL   $D0A1
D099   A91D     LDA   #$1D
D09B   4C9CD2   JMP   $D29C
D09E   18       CLC
D09F   690B     ADC   #$0B
;各板名称简写的处理
D0A1   AA       TAX
D0A2   38       SEC
D0A3   E902     SBC   #$02
D0A5   8588     STA   $88
D0A7   BD05CD   LDA   $CD05,X
D0AA   4CB2D0   JMP   $D0B2
;输入 OK 的处理
D0AD   A643     LDX   $43
D0AF   BD16CD   LDA   $CD16,X
D0B2   8587     STA   $87
D0B4   20DDD0   JSR   $D0DD
D0B7   A91B     LDA   #$1B
D0B9   4C9CD2   JMP   $D29C
D0BC   60       RTS
;输入 NO 的处理
D0BD   A91E     LDA   #$1E
D0BF   209CD2   JSR   $D29C
D0C2   A907     LDA   #$07
D0C4   8587     STA   $87
D0C6   E643     INC   $43
D0C8   A543     LDA   $43
D0CA   C904     CMP   #$04
D0CC   D004     BNE   $D0D2
D0CE   A900     LDA   #$00
D0D0   8543     STA   $43
D0D2   18       CLC
D0D3   690F     ADC   #$0F
D0D5   8588     STA   $88
D0D7   4C13CF   JMP   $CF13
D0DA   20CED0   JSR   $D0CE
D0DD   A206     LDX   #$06
D0DF   2013CF   JSR   $CF13
D0E2   9580     STA   $80,X
D0E4   CA       DEX
D0E5   10FB     BPL   $D0E2
D0E7   60       RTS
D0E8   A92F     LDA   #$2F
D0EA   8587     STA   $87
D0EC   8545     STA   $45
D0EE   A9FF     LDA   #$FF
D0F0   8577     STA   $77
;取功能键定义
D0F2   205DD0   JSR   $D05D
```

```
D0F5    F00B      BEQ    $D102
D0F7    A52D      LDA    $2D
D0F9    D007      BNE    $D102
D0FB    85FC      STA    $FC
D0FD    A92E      LDA    #$2E
D0FF    4C9CD2    JMP    $D29C
D102    60        RTS
D103    A545      LDA    $45
D105    8587      STA    $87
D107    A587      LDA    $87
D109    4C9CD2    JMP    $D29C
D10C    4CBDD0    JMP    $D0BD
    运算板的进入处理
D10F    20FBEE    JSR    $EEFB
D112    A208      LDX    #$08
;定义计算板的功能键
D114    20B3F4    JSR    $F4B3
;从数据库取回本板数据
D117    20FED1    JSR    $D1FE
D11A    A901      LDA    #$01
D11C    A200      LDX    #$00
D11E    8641      STX    $41
D120    8672      STX    $72
D122    F03B      BEQ    $D15F
D124    A93A      LDA    #$3A
D126    8500      STA    $00
D128    A9CA      LDA    #$CA
D12A    2092D1    JSR    $D192
D12D    20FBEE    JSR    $EEFB
D130    200DD2    JSR    $D20D
D133    A902      LDA    #$02
D135    A201      LDX    #$01
D137    D026      BNE    $D15F
    音乐板的进入处理
D139    20FBEE    JSR    $EEFB
;从数据库取回本板数据
D13C    20B0E4    JSR    $E4B0
D13F    A936      LDA    #$36
D141    8500      STA    $00
D143    A9CA      LDA    #$CA
;定义六个卡通
D145    2092D1    JSR    $D192
D148    A210      LDX    #$10
;定义功能键
D14A    20B3F4    JSR    $F4B3
D14D    A9B7      LDA    #$B7
D14F    8531      STA    $31
D151    A202      LDX    #$02
D153    868C      STX    $8C
;定义演奏速度指示器
D155    20D2E6    JSR    $E6D2
;定义三音框
D158    2087E6    JSR    $E687
D15B    A900      LDA    #$00
D15D    A204      LDX    #$04
D15F    85FC      STA    $FC
D161    8636      STX    $36
D163    2083D1    JSR    $D183
D166    2013CF    JSR    $CF13
D169    A040      LDY    #$40
D16B    8479      STY    $79
D16D    999F00    STA    $009F,Y
D170    88        DEY
D171    D0FA      BNE    $D16D
D173    A901      LDA    #$01
D175    85FE      STA    $FE
D177    8D3A70    STA    $703A
D17A    8D3B70    STA    $703B
D17D    200DEF    JSR    $EF0D
D180    4C0FF5    JMP    $F50F
    MUS·、CAL·、MES·三板定义羽毛笔
D183    A93E      LDA    #$3E
D185    8500      STA    $00
D187    A9CA      LDA    #$CA
D189    4C92D1    JMP    $D192
;按 CapsLock 键定义羽毛笔为红色
D18C    A942      LDA    #$42
D18E    8500      STA    $00
D190    A9CA      LDA    #$CA
D192    8501      STA    $01
D194    4C86EC    JMP    $EC86
    Home、Home+Shift 键的处理
D197    A636      LDX    $36
D199    A57D      LDA    $7D
D19B    C9FF      CMP    #$FF
D19D    F041      BEQ    $D1E0
D19F    C901      CMP    #$01
D1A1    D045      BNE    $D1E8
```

```
;按 Home+Shift 键的处理
D1A3   A920     LDA   #$20
D1A5   8D0103   STA   $0301
D1A8   857C     STA   $7C
D1AA   A9E3     LDA   #$E3
D1AC   8D0203   STA   $0302
D1AF   A95A     LDA   #$5A
D1B1   8D0303   STA   $0303
D1B4   A920     LDA   #$20
D1B6   8D0403   STA   $0304
D1B9   A900     LDA   #$00
D1BB   8D0503   STA   $0305
D1BE   20FBEE   JSR   $EEFB
D1C1   A536     LDA   $36
D1C3   C904     CMP   #$04
D1C5   D00B     BNE   $D1D2
D1C7   A57D     LDA   $7D
D1C9   2903     AND   #$03
D1CB   D005     BNE   $D1D2
D1CD   8D0103   STA   $0301
D1D0   F005     BEQ   $D1D7
D1D2   A904     LDA   #$04
D1D4   8D0003   STA   $0300
D1D7   E67D     INC   $7D
D1D9   BD46CA   LDA   $CA46,X
D1DC   C57D     CMP   $7D
D1DE   D007     BNE   $D1E7
;按 Home 键的处理,把羽毛笔移到屏幕左上角
D1E0   200DEF   JSR   $EF0D
D1E3   A900     LDA   #$00
D1E5   857D     STA   $7D
D1E7   60       RTS
D1E8   18       CLC
D1E9   BD4BCA   LDA   $CA4B,X
D1EC   6D0203   ADC   $0302
D1EF   8D0203   STA   $0302
D1F2   A57C     LDA   $7C
D1F4   6900     ADC   #$00
D1F6   8D0103   STA   $0301
D1F9   857C     STA   $7C
D1FB   4CC1D1   JMP   $D1C1
    从数据库送回 CAL·板的原数据
D1FE   2060D2   JSR   $D260
D201   A2D8     LDX   #$D8
D203   BDFF70   LDA   $70FF,X
D206   9D3C06   STA   $063C,X
D209   CA       DEX
D20A   D0F7     BNE   $D203
D20C   60       RTS
    从数据库送回 MES·板的原数据
D20D   2060D2   JSR   $D260
D210   A200     LDX   #$00
D212   BD0074   LDA   $7400,X
D215   9D3D06   STA   $063D,X
D218   E8       INX
D219   D0F7     BNE   $D212
D21B   BD0075   LDA   $7500,X
D21E   9D3D07   STA   $073D,X
D221   E8       INX
D222   E098     CPX   #$98
D224   D0F5     BNE   $D21B
D226   60       RTS
D227   A62F     LDX   $2F
D229   A52E     LDA   $2E
D22B   F006     BEQ   $D233
D22D   BD3D07   LDA   $073D,X
D230   4C36D2   JMP   $D236
D233   BD3D06   LDA   $063D,X
D236   8538     STA   $38
D238   C921     CMP   #$21
D23A   9003     BCC   $D23F
D23C   2088F2   JSR   $F288
D23F   2010F2   JSR   $F210
D242   C902     CMP   #$02
D244   D007     BNE   $D24D
D246   A900     LDA   #$00
D248   8579     STA   $79
D24A   4C0DEF   JMP   $EF0D
D24D   2073F2   JSR   $F273
D250   A636     LDX   $36
D252   E004     CPX   #$04
D254   D003     BNE   $D259
D256   20D4F0   JSR   $F0D4
D259   A538     LDA   $38
D25B   C921     CMP   #$21
D25D   90C8     BCC   $D227
D25F   60       RTS
D260   A95A     LDA   #$5A
```

```
D262    CD3A70    CMP    $703A
D265    D008      BNE    $D26F
D267    AD3B70    LDA    $703B
D26A    C9CC      CMP    #$CC
D26C    D011      BNE    $D27F
D26E    60        RTS
D26F    A920      LDA    #$20
D271    8D0070    STA    $7000
D274    A90D      LDA    #$0D
D276    8D0170    STA    $7001
D279    8D1370    STA    $7013
D27C    8D2670    STA    $7026
;清对话系统数据库
D27F    A000      LDY    #$00
D281    A93C      LDA    #$3C
D283    8508      STA    $08
D285    A970      LDA    #$70
D287    8509      STA    $09
D289    A920      LDA    #$20
D28B    9108      STA    ($08),Y
D28D    E608      INC    $08
D28F    D0F8      BNE    $D289
D291    E609      INC    $09
D293    A509      LDA    $09
D295    C978      CMP    #$78
D297    D0F0      BNE    $D289
D299    60        RTS
D29A    A919      LDA    #$19
D29C    8562      STA    $62
D29E    2013CF    JSR    $CF13
D2A1    8564      STA    $64
D2A3    60        RTS
        对话板主控程序
D2A4    A5C3      LDA    $C3
D2A6    C919      CMP    #$19
D2A8    D004      BNE    $D2AE
D2AA    A900      LDA    #$00
D2AC    8546      STA    $46
D2AE    2009D7    JSR    $D709
D2B1    A5C6      LDA    $C6
D2B3    F00C      BEQ    $D2C1
D2B5    100D      BPL    $D2C4
D2B7    A90A      LDA    #$0A
D2B9    85B7      STA    $B7
```

```
;清 OPERATOR 栏
D2BB    20FED3    JSR    $D3FE
D2BE    4C2BD4    JMP    $D42B
D2C1    4CF1D3    JMP    $D3F1
        COMPUTER 栏送显字符的处理
D2C4    A573      LDA    $73
D2C6    F003      BEQ    $D2CB
D2C8    202BD4    JSR    $D42B
D2CB    A900      LDA    #$00
D2CD    2081D7    JSR    $D781
D2D0    2043D4    JSR    $D443
D2D3    A5C2      LDA    $C2
D2D5    F003      BEQ    $D2DA
D2D7    4CB0D3    JMP    $D3B0
;COMPUTER 栏的叙述送显处理
D2DA    20E0D3    JSR    $D3E0
D2DD    8500      STA    $00
D2DF    A500      LDA    $00
D2E1    48        PHA
D2E2    A900      LDA    #$00
D2E4    2081D7    JSR    $D781
D2E7    68        PLA
D2E8    8500      STA    $00
D2EA    A5BB      LDA    $BB
D2EC    85BA      STA    $BA
D2EE    A5BC      LDA    $BC
D2F0    85B9      STA    $B9
D2F2    E6C5      INC    $C5
D2F4    A500      LDA    $00
D2F6    C90D      CMP    #$0D
D2F8    F060      BEQ    $D35A
D2FA    C9FF      CMP    #$FF
D2FC    F029      BEQ    $D327
D2FE    29E0      AND    #$E0
D300    C9E0      CMP    #$E0
D302    B04C      BCS    $D350
D304    A500      LDA    $00
D306    29C0      AND    #$C0
D308    C9C0      CMP    #$C0
D30A    B072      BCS    $D37E
D30C    A906      LDA    #$06
D30E    8529      STA    $29
D310    A500      LDA    $00
D312    48        PHA
```

```
D313   C920     CMP   #$20
D315   F004     BEQ   $D31B
D317   A902     LDA   #$02
D319   85FF     STA   $FF
D31B   68       PLA
;把显示内容送入三页
D31C   20D9D7   JSR   $D7D9
D31F   A901     LDA   #$01
D321   20A4D3   JSR   $D3A4
D324   4CCFD3   JMP   $D3CF
;每次送显结束时的处理
D327   A001     LDY   #$01
D329   A5C3     LDA   $C3
D32B   C931     CMP   #$31
D32D   F015     BEQ   $D344
D32F   C933     CMP   #$33
D331   F011     BEQ   $D344
D333   C92E     CMP   #$2E
D335   F00B     BEQ   $D342
D337   C936     CMP   #$36
D339   D006     BNE   $D341
D33B   A577     LDA   $77
D33D   C9FF     CMP   #$FF
D33F   D003     BNE   $D344
D341   C8       INY
D342   84FE     STY   $FE
D344   A958     LDA   #$58
D346   852A     STA   $2A
D348   2013CF   JSR   $CF13
D34B   85C6     STA   $C6
D34D   4C75D7   JMP   $D775
D350   A500     LDA   $00
D352   291F     AND   #$1F
D354   20A4D3   JSR   $D3A4
D357   4CCFD3   JMP   $D3CF
;送显换行处理
D35A   A5BB     LDA   $BB
D35C   291F     AND   #$1F
D35E   C907     CMP   #$07
D360   D011     BNE   $D373
D362   C6C5     DEC   $C5
D364   C6C5     DEC   $C5
D366   20E0D3   JSR   $D3E0
D369   E6C5     INC   $C5
D36B   E6C5     INC   $C5
D36D   29F0     AND   #$F0
D36F   C9C0     CMP   #$C0
D371   F03C     BEQ   $D3AF
D373   A5BB     LDA   $BB
D375   29E0     AND   #$E0
D377   85BB     STA   $BB
D379   A927     LDA   #$27
D37B   4CA4D3   JMP   $D3A4
D37E   A588     LDA   $88
D380   A400     LDY   $00
D382   C0C2     CPY   #$C2
D384   F00C     BEQ   $D392
D386   A56B     LDA   $6B
D388   C0C1     CPY   #$C1
D38A   F004     BEQ   $D390
D38C   4A       LSR
D38D   4A       LSR
D38E   4A       LSR
D38F   4A       LSR
D390   290F     AND   #$0F
D392   AA       TAX
D393   BD1ACD   LDA   $CD1A,X
D396   85C0     STA   $C0
D398   BD2ECD   LDA   $CD2E,X
D39B   85C1     STA   $C1
D39D   A900     LDA   #$00
D39F   85C4     STA   $C4
D3A1   E6C2     INC   $C2
D3A3   60       RTS
;PPU 显示地址指针的加一处理
D3A4   18       CLC
D3A5   65BB     ADC   $BB
D3A7   85BB     STA   $BB
D3A9   A5BC     LDA   $BC
D3AB   6900     ADC   #$00
D3AD   85BC     STA   $BC
D3AF   60       RTS
;把操作者输入的内容在 COMPUTER 栏显示
D3B0   2043D4   JSR   $D443
D3B3   A4C4     LDY   $C4
D3B5   B1C0     LDA   ($C0),Y
D3B7   C90D     CMP   #$0D
D3B9   F00F     BEQ   $D3CA
```

```
D3BB    C9BF      CMP   #$BF
D3BD    9002      BCC   $D3C1
D3BF    A920      LDA   #$20
D3C1    E6C4      INC   $C4
D3C3    C6C5      DEC   $C5
D3C5    8500      STA   $00
D3C7    4CDFD2    JMP   $D2DF
D3CA    A900      LDA   #$00
D3CC    85C2      STA   $C2
D3CE    60        RTS
```

COMPUTER 栏换行处理

```
D3CF    A5BB      LDA   $BB
D3D1    291F      AND   #$1F
D3D3    38        SEC
D3D4    E919      SBC   #$19
D3D6    30D7      BMI   $D3AF
D3D8    48        PHA
D3D9    2073D3    JSR   $D373
D3DC    68        PLA
D3DD    4CA4D3    JMP   $D3A4
```

COMPUTER 栏询问内容的选择

```
D3E0    A4C3      LDY   $C3
D3E2    B929C5    LDA   $C529,Y
D3E5    8500      STA   $00
D3E7    B965C5    LDA   $C565,Y
D3EA    8501      STA   $01
D3EC    A4C5      LDY   $C5
D3EE    B100      LDA   ($00),Y
D3F0    60        RTS
D3F1    A5B7      LDA   $B7
D3F3    D009      BNE   $D3FE
D3F5    A52A      LDA   $2A
D3F7    D0D5      BNE   $D3CE
D3F9    E6B7      INC   $B7
D3FB    4C35D6    JMP   $D635
```

清两栏的处理

```
D3FE    A6B7      LDX   $B7
D400    CA        DEX
D401    BDBCC4    LDA   $C4BC,X
D404    85BA      STA   $BA
D406    BDB1C4    LDA   $C4B1,X
D409    85B9      STA   $B9
D40B    E00B      CPX   #$0B
D40D    F012      BEQ   $D421
D40F    A900      LDA   #$00
D411    8D0003    STA   $0300
D414    A920      LDA   #$20
D416    20D9D7    JSR   $D7D9
D419    A954      LDA   #$54
D41B    8D0303    STA   $0303
D41E    E6B7      INC   $B7
D420    60        RTS
```

;赋值子程序

```
D421    202BD4    JSR   $D42B
D424    A930      LDA   #$30
D426    A00D      LDY   #$0D
D428    4CD3D7    JMP   $D7D3
D42B    A901      LDA   #$01
D42D    85C6      STA   $C6
D42F    A9E7      LDA   #$E7
D431    85BB      STA   $BB
D433    A920      LDA   #$20
D435    85BC      STA   $BC
D437    A900      LDA   #$00
D439    85C5      STA   $C5
D43B    85B7      STA   $B7
D43D    A9F8      LDA   #$F8
D43F    8DC002    STA   $02C0
D442    60        RTS
D443    A529      LDA   $29
D445    F002      BEQ   $D449
D447    68        PLA
D448    68        PLA
D449    60        RTS
```

送显功能键提示 END、HELLO、NO、OK 的处理

```
D44A    A573      LDA   $73
D44C    F00E      BEQ   $D45C
D44E    A903      LDA   #$03
D450    85C7      STA   $C7
D452    A56B      LDA   $6B
D454    D004      BNE   $D45A
D456    A004      LDY   #$04
D458    84FE      STY   $FE
D45A    85C8      STA   $C8
D45C    A20A      LDX   #$0A
D45E    A4C7      LDY   $C7
D460    207DD4    JSR   $D47D
```

```
D463    A4C7      LDY    $C7
D465    208FD4    JSR    $D48F
D468    20FBEE    JSR    $EEFB
D46B    C6C7      DEC    $C7
D46D    100D      BPL    $D47C
D46F    2013CF    JSR    $CF13
D472    A5C8      LDA    $C8
D474    C902      CMP    #$02
D476    D004      BNE    $D47C
D478    A904      LDA    #$04
D47A    85C8      STA    $C8
D47C    60        RTS
D47D    B964CA    LDA    $CA64,Y
D480    48        PHA
D481    2044D6    JSR    $D644
D484    68        PLA
D485    8500      STA    $00
D487    A900      LDA    #$00
D489    8D0003    STA    $0300
D48C    4C65ED    JMP    $ED65
```

决定功能键提示栏操作种类的处理

```
D48F    A6C8      LDX    $C8
D491    BD50CA    LDA    $CA50,X
D494    8501      STA    $01
D496    BD54CA    LDA    $CA54,X
D499    8502      STA    $02
D49B    B101      LDA    ($01),Y
D49D    AA        TAX
D49E    BD8DCA    LDA    $CA8D,X
D4A1    8501      STA    $01
D4A3    BDA2CA    LDA    $CAA2,X
D4A6    8502      STA    $02
D4A8    A000      LDY    #$00
D4AA    B101      LDA    ($01),Y
D4AC    C90D      CMP    #$0D
D4AE    F0CC      BEQ    $D47C
D4B0    990403    STA    $0304,Y
D4B3    993D06    STA    $063D,Y
D4B6    C8        INY
D4B7    4CAAD4    JMP    $D4AA
```

功能键的判断处理

```
D4BA    A537      LDA    $37
D4BC    C9C1      CMP    #$C1
D4BE    9054      BCC    $D514
D4C0    C9C5      CMP    #$C5
D4C2    B050      BCS    $D514
D4C4    48        PHA
D4C5    A004      LDY    #$04
D4C7    A20B      LDX    #$0B
D4C9    207DD4    JSR    $D47D
D4CC    68        PLA
D4CD    290F      AND    #$0F
D4CF    A8        TAY
D4D0    88        DEY
D4D1    208FD4    JSR    $D48F
D4D4    C000      CPY    #$00
D4D6    F034      BEQ    $D50C
D4D8    840F      STY    $0F
D4DA    20FBEE    JSR    $EEFB
D4DD    A220      LDX    #$20
D4DF    A920      LDA    #$20
D4E1    9D0303    STA    $0303,X
D4E4    CA        DEX
D4E5    E40F      CPX    $0F
D4E7    D0F6      BNE    $D4DF
D4E9    BD0403    LDA    $0304,X
D4EC    9D3D06    STA    $063D,X
D4EF    2010F2    JSR    $F210
D4F2    2015F1    JSR    $F115
D4F5    CA        DEX
D4F6    10F1      BPL    $D4E9
D4F8    A915      LDA    #$15
D4FA    8D0003    STA    $0300
D4FD    A912      LDA    #$12
D4FF    8D0303    STA    $0303
D502    A900      LDA    #$00
D504    8D1603    STA    $0316
D507    A940      LDA    #$40
D509    85FE      STA    $FE
D50B    60        RTS
D50C    A900      LDA    #$00
D50E    8D0003    STA    $0300
D511    8D0103    STA    $0301
D514    60        RTS
```

复位画面中各卡通的定义

```
D515    A903      LDA    #$03
D517    8536      STA    $36
D519    200DEF    JSR    $EF0D
```

```
;改变卡通状态和 PPU 配色
D51C    2009D7    JSR    $D709
D51F    A56B      LDA    $6B
D521    C903      CMP    #$03
D523    F00A      BEQ    $D52F
D525    A573      LDA    $73
D527    F003      BEQ    $D52C
;定义卡通块显示阴体字 COMPUTER、
OPERATOR 及信号灯等
D529    2035D5    JSR    $D535
D52C    4C73D5    JMP    $D573
D52F    2035D5    JSR    $D535
D532    8560      STA    $60
D534    60        RTS
D535    207CD6    JSR    $D67C
D538    A901      LDA    #$01
D53A    85B6      STA    $B6
D53C    A900      LDA    #$00
D53E    85B7      STA    $B7
D540    85C2      STA    $C2
D542    60        RTS
D543    A5B7      LDA    $B7
D545    D006      BNE    $D54D
D547    A9C6      LDA    #$C6
D549    8551      STA    $51
D54B    85B7      STA    $B7
D54D    C9DA      CMP    #$DA
D54F    F019      BEQ    $D56A
D551    A020      LDY    #$20
D553    20E8ED    JSR    $EDE8
D556    A552      LDA    $52
D558    2903      AND    #$03
D55A    AA        TAX
D55B    BD25C5    LDA    $C525,X
D55E    995003    STA    $0350,Y
D561    88        DEY
D562    D0EF      BNE    $D553
D564    206DD7    JSR    $D76D
D567    4C1CD6    JMP    $D61C
D56A    A900      LDA    #$00
D56C    85B7      STA    $B7
D56E    852C      STA    $2C
D570    E6B6      INC    $B6
D572    60        RTS
;绘制复位画面时,根据
$B6的取值控制程序转向
D573    A6B6      LDX    $B6
D575    CA        DEX
D576    CA        DEX
D577    CA        DEX
D578    F05E      BEQ    $D5D8
D57A    305C      BMI    $D5D8
D57C    CA        DEX
D57D    F0C4      BEQ    $D543
D57F    CA        DEX
D580    F007      BEQ    $D589
D582    CA        DEX
D583    F00C      BEQ    $D591
D585    CA        DEX
D586    F028      BEQ    $D5B0
D588    60        RTS
;把 COMPUTER 栏填满方块的处理
D589    A204      LDX    #$04
D58B    203ED6    JSR    $D63E
D58E    E6B6      INC    $B6
D590    60        RTS
D591    A537      LDA    $37
D593    F00B      BEQ    $D5A0
D595    E6B6      INC    $B6
D597    A904      LDA    #$04
D599    85FE      STA    $FE
D59B    A900      LDA    #$00
D59D    85FC      STA    $FC
D59F    60        RTS
D5A0    A553      LDA    $53
D5A2    2907      AND    #$07
D5A4    AA        TAX
D5A5    BD1DC5    LDA    $C51D,X
D5A8    85B7      STA    $B7
D5AA    20B9D5    JSR    $D5B9
D5AD    4C43D5    JMP    $D543
D5B0    2035D6    JSR    $D635
D5B3    2013CF    JSR    $CF13
D5B6    85B7      STA    $B7
D5B8    60        RTS
    复位画面中的发声处理
D5B9    A200      LDX    #$00
D5BB    A52C      LDA    $2C
```

```
D5BD   D005     BNE   $D5C4
D5BF   C62C     DEC   $2C
D5C1   4CD1D5   JMP   $D5D1
D5C4   C9C8     CMP   #$C8
D5C6   F00A     BEQ   $D5D2
D5C8   C9D0     CMP   #$D0
D5CA   F005     BEQ   $D5D1
D5CC   C9F0     CMP   #$F0
D5CE   D007     BNE   $D5D7
D5D0   E8       INX
D5D1   E8       INX
D5D2   BDAEC4   LDA   $C4AE,X
D5D5   85FC     STA   $FC
D5D7   60       RTS
```

复位画面中对操作者输入栏的显示处理

```
D5D8   E8       INX
D5D9   F010     BEQ   $D5EB
D5DB   E8       INX
D5DC   F01B     BEQ   $D5F9
D5DE   A921     LDA   #$21
D5E0   A007     LDY   #$07
D5E2   20D3D7   JSR   $D7D3
D5E5   203CD6   JSR   $D63C
D5E8   E6B6     INC   $B6
D5EA   60       RTS
```

复位画面中对功能键提示栏的显示处理

```
D5EB   A20C     LDX   #$0C
D5ED   A4C7     LDY   $C7
D5EF   207DD4   JSR   $D47D
D5F2   C6C7     DEC   $C7
D5F4   1002     BPL   $D5F8
D5F6   E6B6     INC   $B6
D5F8   60       RTS
D5F9   A5B7     LDA   $B7
D5FB   D004     BNE   $D601
D5FD   A909     LDA   #$09
D5FF   85B7     STA   $B7
D601   A900     LDA   #$00
D603   8586     STA   $86
D605   85C6     STA   $C6
D607   20FED3   JSR   $D3FE
D60A   A9FF     LDA   #$FF
D60C   8D0403   STA   $0304
D60F   A5C6     LDA   $C6
D611   F0E5     BEQ   $D5F8
D613   E6B6     INC   $B6
D615   A903     LDA   #$03
D617   85C7     STA   $C7
D619   4C2BD4   JMP   $D42B
```

复位画面中彩色块的送显控制处理

```
D61C   A984     LDA   #$84
D61E   8D5003   STA   $0350
D621   A208     LDX   #$08
D623   2044D6   JSR   $D644
D626   A5B7     LDA   $B7
D628   8500     STA   $00
D62A   2065ED   JSR   $ED65
D62D   A5B7     LDA   $B7
D62F   18       CLC
D630   6904     ADC   #$04
D632   85B7     STA   $B7
D634   60       RTS
D635   A90F     LDA   #$0F
D637   A007     LDY   #$07
D639   20D3D7   JSR   $D7D3
D63C   A205     LDX   #$05
D63E   2044D6   JSR   $D644
D641   4C65ED   JMP   $ED65
D644   BDC7C4   LDA   $C4C7,X
D647   8505     STA   $05
D649   BDD4C4   LDA   $C4D4,X
D64C   8506     STA   $06
D64E   A004     LDY   #$04
D650   B105     LDA   ($05),Y
D652   990000   STA   $0000,Y
D655   88       DEY
D656   10F8     BPL   $D650
D658   60       RTS
```

改变 PPU 的配色处理

```
D659   A522     LDA   $22
D65B   D0FB     BNE   $D658
D65D   A53B     LDA   $3B
D65F   2907     AND   #$07
D661   C901     CMP   #$01
D663   D0F3     BNE   $D658
D665   E6B8     INC   $B8
D667   A5B8     LDA   $B8
D669   2907     AND   #$07
```

```
D66B    85B8      STA    $B8
D66D    AA        TAX
D66E    D004      BNE    $D674
D670    A930      LDA    #$30
D672    8522      STA    $22
D674    BDA6C4    LDA    $C4A6,X
D677    A003      LDY    #$03
D679    4CD3D7    JMP    $D7D3
D67C    208BD6    JSR    $D68B
D67F    A26F      LDX    #$6F
D681    20B1D6    JSR    $D6B1
D684    9D4002    STA    $0240,X
D687    CA        DEX
D688    10F7      BPL    $D681
D68A    60        RTS
```

复位画面中各卡通块的定义处理

```
D68B    A22F      LDX    #$2F
D68D    A900      LDA    #$00
D68F    9D0002    STA    $0200,X
D692    CA        DEX
D693    10F8      BPL    $D68D
D695    A206      LDX    #$06
D697    2044D6    JSR    $D644
D69A    20BEEC    JSR    $ECBE
D69D    A207      LDX    #$07
D69F    2044D6    JSR    $D644
D6A2    20BEEC    JSR    $ECBE
D6A5    A20F      LDX    #$0F
D6A7    BD78C4    LDA    $C478,X
D6AA    9D3002    STA    $0230,X
D6AD    CA        DEX
D6AE    10F7      BPL    $D6A7
D6B0    60        RTS
D6B1    A000      LDY    #$00
D6B3    8A        TXA
D6B4    C920      CMP    #$20
D6B6    9017      BCC    $D6CF
D6B8    2903      AND    #$03
D6BA    F030      BEQ    $D6EC
D6BC    C901      CMP    #$01
D6BE    F039      BEQ    $D6F9
D6C0    C903      CMP    #$03
D6C2    D03D      BNE    $D701
D6C4    E050      CPX    #$50
D6C6    9039      BCC    $D701
D6C8    E060      CPX    #$60
D6CA    902E      BCC    $D6FA
D6CC    4CF8D6    JMP    $D6F8
D6CF    2903      AND    #$03
D6D1    F011      BEQ    $D6E4
D6D3    C901      CMP    #$01
D6D5    F028      BEQ    $D6FF
D6D7    C902      CMP    #$02
D6D9    F025      BEQ    $D700
D6DB    8A        TXA
D6DC    290F      AND    #$0F
D6DE    4A        LSR
D6DF    4A        LSR
D6E0    A8        TAY
D6E1    4CFED6    JMP    $D6FE
D6E4    8A        TXA
D6E5    29F0      AND    #$F0
D6E7    F01A      BEQ    $D703
D6E9    4C02D7    JMP    $D702
D6EC    8A        TXA
D6ED    C950      CMP    #$50
D6EF    B010      BCS    $D701
D6F1    2908      AND    #$08
D6F3    D010      BNE    $D705
D6F5    4C04D7    JMP    $D704
D6F8    C8        INY
D6F9    C8        INY
D6FA    C8        INY
D6FB    C8        INY
D6FC    C8        INY
D6FD    C8        INY
D6FE    C8        INY
D6FF    C8        INY
D700    C8        INY
D701    C8        INY
D702    C8        INY
D703    C8        INY
D704    C8        INY
D705    B988C4    LDA    $C488,Y
D708    60        RTS
```

复位画面中控制各卡通块的状态

改变子程序

```
D709    207BD7    JSR    $D77B
```

```
D70C   206DD7   JSR   $D76D
D70F   2059D6   JSR   $D659
D712   A53B     LDA   $3B
D714   2903     AND   #$03
D716   C901     CMP   #$01
D718   D052     BNE   $D76C
D71A   A204     LDX   #$04
D71C   B551     LDA   $51,X
D71E   9D5003   STA   $0350,X
D721   CA       DEX
D722   D0F8     BNE   $D71C
D724   A914     LDA   #$14
D726   8D5003   STA   $0350
D729   A209     LDX   #$09
D72B   203ED6   JSR   $D63E
D72E   A5B0     LDA   $B0
D730   25B1     AND   $B1
D732   25B2     AND   $B2
D734   25B3     AND   $B3
D736   F007     BEQ   $D73F
D738   A00B     LDY   #$0B
D73A   A551     LDA   $51
D73C   20D3D7   JSR   $D7D3
D73F   A203     LDX   #$03
D741   B525     LDA   $25,X
D743   D018     BNE   $D75D
D745   B551     LDA   $51,X
D747   2937     AND   #$37
D749   9525     STA   $25,X
D74B   B5B0     LDA   $B0,X
D74D   4901     EOR   #$01
D74F   95B0     STA   $B0,X
D751   B5B0     LDA   $B0,X
D753   09F0     ORA   #$F0
D755   8500     STA   $00
D757   BD9EC4   LDA   $C49E,X
D75A   2092D7   JSR   $D792
D75D   B5B0     LDA   $B0,X
D75F   F008     BEQ   $D769
D761   8600     STX   $00
D763   BD9AC4   LDA   $C49A,X
D766   2092D7   JSR   $D792
D769   CA       DEX
D76A   10D5     BPL   $D741
D76C   60       RTS
D76D   AD0003   LDA   $0300
D770   F002     BEQ   $D774
D772   68       PLA
D773   68       PLA
D774   60       RTS
D775   A8       TAY
D776   A9F0     LDA   #$F0
D778   4C8AD7   JMP   $D78A
```

;把卡通的状态参数写入二页

```
D77B   A910     LDA   #$10
D77D   A424     LDY   $24
D77F   F0F4     BEQ   $D775
D781   A8       TAY
D782   A53B     LDA   $3B
D784   2902     AND   #$02
D786   F0EC     BEQ   $D774
D788   A904     LDA   #$04
D78A   8500     STA   $00
D78C   A200     LDX   #$00
D78E   98       TYA
D78F   18       CLC
D790   6942     ADC   #$42
D792   BCA2C4   LDY   $C4A2,X
D795   8503     STA   $03
D797   A902     LDA   #$02
D799   8504     STA   $04
D79B   8A       TXA
D79C   48       PHA
D79D   A200     LDX   #$00
D79F   8406     STY   $06
D7A1   A500     LDA   $00
D7A3   3015     BMI   $D7BA
D7A5   C904     CMP   #$04
D7A7   F017     BEQ   $D7C0
D7A9   B551     LDA   $51,X
D7AB   2918     AND   #$18
D7AD   860F     STX   $0F
D7AF   A600     LDX   $00
D7B1   18       CLC
D7B2   7D96C4   ADC   $C496,X
D7B5   A60F     LDX   $0F
D7B7   4CC5D7   JMP   $D7C5
D7BA   A002     LDY   #$02
```

```
D7BC   290F     AND   #$0F
D7BE   F002     BEQ   $D7C2
D7C0   B451     LDY   $51,X
D7C2   98       TYA
D7C3   2903     AND   #$03
D7C5   A406     LDY   $06
D7C7   9103     STA   ($03),Y
D7C9   E8       INX
D7CA   88       DEY
D7CB   88       DEY
D7CC   88       DEY
D7CD   88       DEY
D7CE   10CF     BPL   $D79F
D7D0   68       PLA
D7D1   AA       TAX
D7D2   60       RTS
D7D3   84BA     STY   $BA
D7D5   A03F     LDY   #$3F
D7D7   84B9     STY   $B9
D7D9   8D9103   STA   $0391
D7DC   A911     LDA   #$11
D7DE   8D9003   STA   $0390
D7E1   A5BA     LDA   $BA
D7E3   8500     STA   $00
D7E5   A5B9     LDA   $B9
D7E7   8501     STA   $01
D7E9   A990     LDA   #$90
D7EB   8502     STA   $02
D7ED   A903     LDA   #$03
D7EF   8503     STA   $03
D7F1   4C65ED   JMP   $ED65
```

复位画面、各板画面的绘制数据

下面划双线的为送数指南，划单线的为数据

```
D7F4                3F 00 20 0F 30 00 0F 0F 30 00 0F 0F
D800  30 00 0F 0F 30 0F 0F 0F 30 27 0F 0F 30 02 0F 0F
D810  30 16 0F 0F 30 2A 0F 23 C1 46 AA 23 C9 06 C0 F0
D820  F0 F0 F0 30 23 D1 06 CC FF FF FF FF 33 23 D9 06
D830  0C 0F 0F 0F 0F 03 23 E1 06 44 55 55 55 55 11 23
D840  E9 06 44 55 55 55 55 11 23 F0 50 AA 23 F1 46 55
D850  20 00 DE FE 20 01 DE 16 20 1E DE 02 20 1F DE FE
D860  20 BF CA 1E 22 3F C5 1E 20 02 DE 02 20 1D DE 16
D870  20 07 52 FE 20 27 52 FE 20 47 52 FE 20 06 C3 16
D880  20 19 C3 02 20 27 44 1C 20 2E 44 1C 20 35 44 1C
D890  20 47 44 1C 20 4E 44 1C 20 55 44 1C 20 A1 CA 01
D8A0  20 A2 CA 00 20 BD CA 01 20 BE CA 00 21 E1 02 1D
D8B0  1F 21 FD 02 1D 1F 20 C3 88 03 04 05 20 20 03 04
D8C0  05 20 E4 01 06 21 84 01 06 20 85 8C 09 0A 07 08
D8D0  07 07 07 07 08 07 0B 0C 20 86 54 0D 20 A6 54 0E
D8E0  21 C6 54 0F 21 E6 54 10 20 9A 8C 11 12 13 14 13
D8F0  13 13 13 14 13 15 18 20 FB 01 17 21 9B 01 17 20
D900  DC 88 19 1A 1B 20 20 19 1A 1B 22 21 C5 01 22 22
D910  C5 00 22 3D C5 01 22 3E C5 00 22 C1 02 1D 1F 22
D920  DD 02 1D 1F 22 43 83 03 04 05 22 64 01 06 22 05
D930  87 09 0A 07 08 07 0B 0C 22 5C 83 19 1A 1B 22 7B
D940  01 17 22 1A 87 11 12 13 14 13 15 18 22 06 54 0D
D950  22 26 54 0E 22 A6 54 0F 22 C6 54 10 23 02 C5 1C
D960  23 1D C5 1C 23 24 58 BC 23 64 58 BD 23 23 83 B9
D970  BA BB 23 29 83 B9 BA BB 23 2F 83 B9 BA BB 23 30
D980  83 B9 BA BB 23 36 83 B9 BA BB 23 3C 83 B9 BA BB
```

D990 23 05 82 BE BF 23 0B 82 BE BF 23 12 82 BE BF 23

D9A0 18 82 BE BF 00

计算板画面数据

D9A5 3F 00 08 0F 30 2C 02 0F 30 36 26

D9B0 20 A8 10 43 41 4C 43 55 4C 41 54 4F 52 20 42 4F

C A L C U L A T O R B O

D9C0 41 52 44 23 04 16 46 31 3A 2B 20 20 46 32 3A 2D

A R D F 1 : + F 2 : −

D9D0 20 20 46 33 3A B5 20 20 46 34 3A B6 23 24 16 46

F 3 : × F 4 : ÷ F

D9E0 35 3A 28 20 20 46 36 3A 29 20 20 46 37 3A 2E 20

5 : (F 6 :) F 7 : .

D9F0 20 46 38 3A 3D 23 D1 5E 55 00

F 8 : =

音乐、打字板部分画面数据

D9FA 3F 00 08 0F 30 27

DA00 07 0F 30 36 15 3F 18 04 0F 36 16 02 20 A9 0E 4D

M

DA10 45 53 53 41 47 45 20 20 42 4F 41 52 44 23 D1 5E

E S S A G E B O A R D

DA20 55 00 21 04 58 B8 21 84 58 B8 22 04 58 B8 22 84

DA30 58 B8 21 44 58 B8 21 C4 58 B8 22 44 58 B8 22 C4

DA40 58 B8 3F 08 10 0F 30 36 07 0F 30 2C 02 0F 26 30

DA50 15 0F 29 30 1A 23 C9 46 55 23 D7 42 00 23 DF 42

DA60 00 23 E7 42 00 23 F1 45 05 23 F6 01 0D 23 1A 02

DA70 F9 FB 23 3A 02 FA FC 20 88 50 ED 20 C8 50 ED 20

DA80 87 83 EF EE F1 20 98 83 F0 EE F2 20 63 5A F6 23

DA90 43 5A F6 20 82 D6 F4 20 9D D6 F4 20 62 01 F3 20

DAA0 7D 01 F7 23 42 01 F5 23 5D 01 F8 20 00 7F FF 20

DAB0 3F 61 FF 23 60 7F FF 23 9F 61 FF 20 60 D8 FF 20

DAC0 61 D8 FF 20 7E D8 FF 20 7F D8 FF 00

音乐板部分画面数据

DACC 3F 00 08 0F

DAD0 30 15 05 0F 30 2C 11 20 AA 0C 4D 55 53 49 43 20

M U S I C

DAE0 20 42 4F 41 52 44 22 E8 05 46 41 53 54 3C 22 ED

B O A R D F A S T <

DAF0 46 2D 22 F3 05 3E 53 4C 4F 57 23 04 15 46 31 2D

> S L O W F 1 −

DB00 46 37 3A 63 64 65 66 67 61 62 20 46 38 3A 50 4C

F 7 : c d e f g a b F 8 : P L

DB10 41 59 23 25 15 28 23 29 20 3A 43 44 45 46 47 41

A Y (#) : C D E F G A

```
DB20  42 20 24 3A 52 45 50 45 41 54 23 D1 5E 99 00 00
       B     $  :  R E P E  A T
```

打字板定义卡通的数据

```
DB30  04 08 04 40 44 40 44 58 5C 58 5C 60 64 60 64 1C
DB40  20 24 1C B8 BC B8 BC C0 C4 C0 C4 C8 CC C8 CC 2F
DB50  33 37 3B 3F 43 47 4B DB DB DB DB DB DB DB DB 14
DB60  0F 36 16 02 14 0F 30 15 12 14 0F 30 12 16 14 0F
DB70  30 26 12 14 0F 35 25 17 14 0F 30 26 19 14 0F 30
DB80  02 16 14 0F 30 26 12 5F 64 69 6E 73 78 7D 82 DB
DB90  DB DB DB DB DB DB DB
```

打字板工作主控程序

```
DB97   A579     LDA   $79
DB99   D026     BNE   $DBC1
DB9B   20E2DB   JSR   $DBE2
DB9E   A57D     LDA   $7D
DBA0   D019     BNE   $DBBB
DBA2   A537     LDA   $37
DBA4   C90B     CMP   #$0B
DBA6   F01C     BEQ   $DBC4
DBA8   C90C     CMP   #$0C
DBAA   F01D     BEQ   $DBC9
DBAC   C91B     CMP   #$1B
DBAE   D006     BNE   $DBB6
;保存打字板内容
DBB0   20CDDB   JSR   $DBCD
DBB3   4C9BDC   JMP   $DC9B
DBB6   A201     LDX   #$01
DBB8   4C2CEF   JMP   $EF2C
DBBB   2097D1   JSR   $D197
DBBE   4C0FF5   JMP   $F50F
DBC1   4C27D2   JMP   $D227
DBC4   A9FF     LDA   #$FF
DBC6   857D     STA   $7D
DBC8   60       RTS
DBC9   A901     LDA   #$01
DBCB   D0F9     BNE   $DBC6
DBCD   A200     LDX   #$00
DBCF   BD3D06   LDA   $063D,X
DBD2   9D0074   STA   $7400,X
DBD5   E8       INX
DBD6   D0F7     BNE   $DBCF
DBD8   BD3D07   LDA   $073D,X
DBDB   9D0075   STA   $7500,X
DBDE   E8       INX
DBDF   D0F7     BNE   $DBD8
DBE1   60       RTS
```

打字板卡通运动控制程序

```
DBE2   A539     LDA   $39
DBE4   F002     BEQ   $DBE8
DBE6   E6B6     INC   $B6
DBE8   A528     LDA   $28
DBEA   F01A     BEQ   $DC06
DBEC   A5B4     LDA   $B4
DBEE   D00D     BNE   $DBFD
DBF0   A908     LDA   #$08
DBF2   85B4     STA   $B4
DBF4   A5B2     LDA   $B2
DBF6   18       CLC
DBF7   6901     ADC   #$01
DBF9   2903     AND   #$03
DBFB   85B2     STA   $B2
DBFD   C6B4     DEC   $B4
DBFF   A5B4     LDA   $B4
DC01   2903     AND   #$03
DC03   F00C     BEQ   $DC11
DC05   60       RTS
DC06   A5B6     LDA   $B6
DC08   F006     BEQ   $DC10
DC0A   A910     LDA   #$10
DC0C   8528     STA   $28
DC0E   C6B6     DEC   $B6
DC10   60       RTS
;MES·板卡通前进的控制处理
DC11   A5B3     LDA   $B3
DC13   38       SEC
DC14   E901     SBC   #$01
DC16   85B3     STA   $B3
```

```
DC18    A5B3      LDA    $B3
DC1A    D00D      BNE    $DC29
DC1C    A5B5      LDA    $B5
DC1E    18        CLC
DC1F    6901      ADC    #$01
DC21    2907      AND    #$07
DC23    85B5      STA    $B5
DC25    A8        TAY
DC26    204EDC    JSR    $DC4E
DC29    A5B3      LDA    $B3
DC2B    8500      STA    $00
DC2D    A9BF      LDA    #$BF
DC2F    8501      STA    $01
DC31    A4B5      LDY    $B5
DC33    B94FDB    LDA    $DB4F,Y
DC36    8503      STA    $03
DC38    B957DB    LDA    $DB57,Y
DC3B    8504      STA    $04
DC3D    A4B2      LDY    $B2
DC3F    B103      LDA    ($03),Y
DC41    8502      STA    $02
DC43    A922      LDA    #$22
DC45    8503      STA    $03
DC47    A900      LDA    #$00
DC49    8504      STA    $04
DC4B    4CBEEC    JMP    $ECBE
DC4E    A918      LDA    #$18
DC50    8500      STA    $00
DC52    A93F      LDA    #$3F
DC54    8501      STA    $01
DC56    B987DB    LDA    $DB87,Y
DC59    8502      STA    $02
DC5B    B98FDB    LDA    $DB8F,Y
DC5E    8503      STA    $03
DC60    4C65ED    JMP    $ED65
```

运算板出错提示数据

```
DC63         57 52 4F 4E 47 0D 4F 56 45 52 46 4C 4F
              W  R  O  N  G     O  V  E  R  F  L  O
DC70    57 0D DC 63 DC 69 DC
        W
```

CALCULATOR(计算板)工作主控程序

```
DC77    A579      LDA    $79
DC79    D071      BNE    $DCEC
DC7B    A57D      LDA    $7D
DC7D    D062      BNE    $DCE1
DC7F    A5D8      LDA    $D8
DC81    D044      BNE    $DCC7
DC83    A541      LDA    $41
DC85    D03F      BNE    $DCC6
DC87    A537      LDA    $37
DC89    C90B      CMP    #$0B
;按 Home 键转 $DCF0
DC8B    F063      BEQ    $DCF0
DC8D    C90C      CMP    #$0C
;按 Home+Shift 键转 $DCF5
DC8F    F064      BEQ    $DCF5
DC91    C91B      CMP    #$1B
DC93    D01B      BNE    $DCB0
;按 ESC 键的处理
DC95    20B1F4    JSR    $F4B1
DC98    209ADF    JSR    $DF9A
DC9B    A95A      LDA    #$5A
DC9D    8D3A70    STA    $703A
DCA0    A9CC      LDA    #$CC
DCA2    8D3B70    STA    $703B
DCA5    A900      LDA    #$00
DCA7    85FC      STA    $FC
DCA9    A901      LDA    #$01
DCAB    85FE      STA    $FE
DCAD    4C13CF    JMP    $CF13
;CAL.板输入字符的处理
DCB0    A200      LDX    #$00
DCB2    202CEF    JSR    $EF2C
DCB5    A5B0      LDA    $B0
DCB7    F006      BEQ    $DCBF
DCB9    205ADE    JSR    $DE5A
DCBC    8541      STA    $41
DCBE    60        RTS
DCBF    A53A      LDA    $3A
DCC1    F003      BEQ    $DCC6
DCC3    2073F4    JSR    $F473
```

```
DCC6    60        RTS
;CAL.板送显运算结果的处理
DCC7    A5D8      LDA     $D8
DCC9    C901      CMP     #$01
DCCB    D004      BNE     $DCD1
DCCD    2031DE    JSR     $DE31
DCD0    60        RTS
DCD1    C902      CMP     #$02
DCD3    D004      BNE     $DCD9
DCD5    20F9DC    JSR     $DCF9
DCD8    60        RTS
DCD9    C903      CMP     #$03
DCDB    D007      BNE     $DCE4
DCDD    2075DD    JSR     $DD75
DCE0    60        RTS
DCE1    2097D1    JSR     $D197
DCE4    200FF5    JSR     $F50F
DCE7    A900      LDA     #$00
DCE9    85D8      STA     $D8
DCEB    60        RTS
DCEC    2027D2    JSR     $D227
DCEF    60        RTS
DCF0    A9FF      LDA     #$FF
DCF2    857D      STA     $7D
DCF4    60        RTS
DCF5    A901      LDA     #$01
DCF7    D0F9      BNE     $DCF2
;CAL.板送显运算结果子程序
DCF9    A521      LDA     $21
DCFB    F001      BEQ     $DCFE
DCFD    60        RTS
DCFE    A908      LDA     #$08
DD00    8521      STA     $21
DD02    A6DA      LDX     $DA
DD04    BDF003    LDA     $03F0,X
DD07    85DD      STA     $DD
DD09    85DE      STA     $DE
DD0B    C90D      CMP     #$0D
DD0D    D036      BNE     $DD45
DD0F    A52F      LDA     $2F
DD11    C9BF      CMP     #$BF
DD13    B00D      BCS     $DD22
DD15    C5B6      CMP     $B6
DD17    F009      BEQ     $DD22
DD19    2010F2    JSR     $F210
DD1C    2015F1    JSR     $F115
DD1F    4C0FDD    JMP     $DD0F
;运算结果送显完毕的处理
DD22    A920      LDA     #$20
DD24    85DD      STA     $DD
DD26    A6B6      LDX     $B6
DD28    BD3D06    LDA     $063D,X
DD2B    C920      CMP     #$20
DD2D    D004      BNE     $DD33
DD2F    A900      LDA     #$00
DD31    85D9      STA     $D9
DD33    2045DD    JSR     $DD45
DD36    A903      LDA     #$03
DD38    85D8      STA     $D8
DD3A    A900      LDA     #$00
DD3C    85DA      STA     $DA
DD3E    C6D9      DEC     $D9
DD40    A940      LDA     #$40
DD42    85FF      STA     $FF
DD44    60        RTS
          送显运算结果
DD45    20F8F3    JSR     $F3F8
DD48    2010F2    JSR     $F210
DD4B    2017F4    JSR     $F417
DD4E    2015F1    JSR     $F115
DD51    A5DD      LDA     $DD
DD53    8D0403    STA     $0304
DD56    A5D9      LDA     $D9
DD58    F002      BEQ     $DD5C
DD5A    E6D9      INC     $D9
DD5C    A4B6      LDY     $B6
DD5E    B93D06    LDA     $063D,Y
DD61    C920      CMP     #$20
DD63    D002      BNE     $DD67
DD65    E6D9      INC     $D9
DD67    A5DD      LDA     $DD
DD69    993D06    STA     $063D,Y
DD6C    E6B6      INC     $B6
DD6E    E6DA      INC     $DA
DD70    A908      LDA     #$08
DD72    85FF      STA     $FF
DD74    60        RTS
DD75    A5D9      LDA     $D9
```

```
DD77   F02B      BEQ   $DDA4
DD79   A6B6      LDX   $B6
DD7B   BD3D06    LDA   $063D,X
DD7E   C920      CMP   #$20
DD80   F019      BEQ   $DD9B
DD82   A920      LDA   #$20
DD84   9D3D06    STA   $063D,X
DD87   20F8F3    JSR   $F3F8
DD8A   2017F4    JSR   $F417
DD8D   A920      LDA   #$20
DD8F   8D0403    STA   $0304
DD92   E6B6      INC   $B6
DD94   A6B6      LDX   $B6
DD96   E0D7      CPX.  #$D7
DD98   B001      BCS   $DD9B
DD9A   60        RTS
DD9B   A920      LDA   #$20
DD9D   9D3D06    STA   $063D,X
DDA0   A900      LDA   #$00
DDA2   85D9      STA   $D9
DDA4   A904      LDA   #$04
DDA6   85D8      STA   $D8
DDA8   60        RTS
```

把存在$430～$43A中的运算结果送入$3F0～$3FA

```
DDA9   A200      LDX   #$00
DDAB   A920      LDA   #$20
DDAD   9DF003    STA   $03F0,X
DDB0   E8        INX
DDB1   E010      CPX   #$10
DDB3   90F6      BCC   $DDAB
DDB5   A200      LDX   #$00
DDB7   A000      LDY   #$00
DDB9   AD3A04    LDA   $043A
DDBC   C920      CMP   #$20
DDBE   F006      BEQ   $DDC6
DDC0   A92D      LDA   #$2D
DDC2   99F003    STA   $03F0,Y
DDC5   C8        INY
DDC6   BD3004    LDA   $0430,X
DDC9   C930      CMP   #$30
DDCB   D012      BNE   $DDDF
DDCD   E8        INX
DDCE   E009      CPX   #$09
DDD0   90F4      BCC   $DDC6
DDD2   A930      LDA   #$30
DDD4   8DF003    STA   $03F0
DDD7   A90D      LDA   #$0D
DDD9   8DF103    STA   $03F1
DDDC   4CFCDD    JMP   $DDFC
DDDF   C92E      CMP   #$2E
DDE1   D008      BNE   $DDEB
DDE3   BD2F04    LDA   $042F,X
DDE6   C930      CMP   #$30
DDE8   D001      BNE   $DDEB
DDEA   CA        DEX
;运算结果的整形处理
DDEB   BD3004    LDA   $0430,X
DDEE   99F003    STA   $03F0,Y
DDF1   E8        INX
DDF2   C8        INY
DDF3   E009      CPX   #$09
DDF5   90F4      BCC   $DDEB
DDF7   A90D      LDA   #$0D
DDF9   99F003    STA   $03F0,Y
DDFC   A901      LDA   #$01
DDFE   85D8      STA   $D8
DE00   60        RTS
```

送显出错提示

```
DE01   0A        ASL
DE02   AA        TAX
DE03   BD71DC    LDA   $DC71,X
DE06   8500      STA   $00
DE08   BD72DC    LDA   $DC72,X
DE0B   8501      STA   $01
DE0D   A200      LDX   #$00
DE0F   A920      LDA   #$20
DE11   9DF003    STA   $03F0,X
DE14   E8        INX
DE15   E010      CPX   #$10
DE17   90F6      BCC   $DE0F
DE19   A000      LDY   #$00
DE1B   B100      LDA   ($00),Y
DE1D   99F003    STA   $03F0,Y
DE20   C90D      CMP   #$0D
DE22   F004      BEQ   $DE28
DE24   C8        INY
DE25   4C1BDE    JMP   $DE1B
```

```
DE28   A901     LDA   #$01
DE2A   85D8     STA   $D8
DE2C   A900     LDA   #$00
DE2E   85B9     STA   $B9
DE30   60       RTS
```

查找算式中的等号，以决定送显运算结果的地址

```
DE31   A9E4     LDA   #$E4
DE33   8535     STA   $35
DE35   A920     LDA   #$20
DE37   8533     STA   $33
DE39   A2FF     LDX   #$FF
DE3B   2017F4   JSR   $F417
DE3E   E8       INX
DE3F   E42F     CPX   $2F
DE41   D0F8     BNE   $DE3B
DE43   BD3D06   LDA   $063D,X
DE46   C93D     CMP   #$3D
DE48   F008     BEQ   $DE52
DE4A   2017F4   JSR   $F417
DE4D   E8       INX
DE4E   E0E0     CPX   #$E0
DE50   90F1     BCC   $DE43
DE52   E8       INX
DE53   86B6     STX   $B6
DE55   A902     LDA   #$02
DE57   85D8     STA   $D8
DE59   60       RTS
```

判断算式是否合法及找出算式的第一个字符

```
DE5A   208FDF   JSR   $DF8F
DE5D   20D2E3   JSR   $E3D2
DE60   A000     LDY   #$00
DE62   A62F     LDX   $2F
DE64   F046     BEQ   $DEAC
DE66   CA       DEX
DE67   BD3D06   LDA   $063D,X
DE6A   C920     CMP   #$20
DE6C   F00B     BEQ   $DE79
DE6E   C93D     CMP   #$3D
DE70   F03A     BEQ   $DEAC
DE72   E000     CPX   #$00
DE74   F004     BEQ   $DE7A
DE76   4C66DE   JMP   $DE66
DE79   E8       INX
DE7A   86C0     STX   $C0
DE7C   BD3D06   LDA   $063D,X
DE7F   990076   STA   $7600,Y
DE82   C93D     CMP   #$3D
DE84   F009     BEQ   $DE8F
DE86   C8       INY
DE87   E8       INX
DE88   E0E0     CPX   #$E0
DE8A   D0F0     BNE   $DE7C
DE8C   A900     LDA   #$00
DE8E   60       RTS
```

;若算式合法则令 $41=1，否则进行出错处理

```
DE8F   A6C0     LDX   $C0
DE91   A900     LDA   #$00
DE93   85BC     STA   $BC
DE95   85BA     STA   $BA
DE97   85BD     STA   $BD
DE99   85B4     STA   $B4
DE9B   85BE     STA   $BE
DE9D   85B5     STA   $B5
DE9F   85B3     STA   $B3
DEA1   85B2     STA   $B2
DEA3   BD3D06   LDA   $063D,X
DEA6   850F     STA   $0F
DEA8   C920     CMP   #$20
DEAA   D003     BNE   $DEAF
DEAC   4C89DF   JMP   $DF89
DEAF   C92E     CMP   #$2E
DEB1   D014     BNE   $DEC7
```

;算式中有小数点的处理

```
DEB3   A5B5     LDA   $B5
DEB5   F003     BEQ   $DEBA
DEB7   4C89DF   JMP   $DF89
DEBA   A5BE     LDA   $BE
DEBC   F0EE     BEQ   $DEAC
DEBE   A901     LDA   #$01
DEC0   85B5     STA   $B5
DEC2   85B3     STA   $B3
DEC4   4C7EDF   JMP   $DF7E
DEC7   C930     CMP   #$30
DEC9   9022     BCC   $DEED
DECB   C93A     CMP   #$3A
```

```
DECD  B01E    BCS  $DEED
DECF  A5BA    LDA  $BA
DED1  D0D9    BNE  $DEAC
DED3  A900    LDA  #$00
DED5  85B3    STA  $B3
DED7  A5BE    LDA  $BE
DED9  D007    BNE  $DEE2
DEDB  85BD    STA  $BD
DEDD  E6BE    INC  $BE
DEDF  4C7EDF  JMP  $DF7E
DEE2  E6BE    INC  $BE
DEE4  A5BE    LDA  $BE
DEE6  C909    CMP  #$09
DEE8  101A    BPL  $DF04
DEEA  4C7EDF  JMP  $DF7E
DEED  C92B    CMP  #$2B
;加法转$DF00
DEEF  F00F    BEQ  $DF00
DEF1  C92D    CMP  #$2D
;减法转$DF00
DEF3  F00B    BEQ  $DF00
DEF5  C9B5    CMP  #$B5
;乘法转$DF00
DEF7  F007    BEQ  $DF00
DEF9  C9B6    CMP  #$B6
;除法转$DF00
DEFB  F003    BEQ  $DF00
DEFD  4C2ADF  JMP  $DF2A
;四则运算处理
DF00  A5BD    LDA  $BD
DF02  F003    BEQ  $DF07
DF04  4C89DF  JMP  $DF89
DF07  A5B3    LDA  $B3
DF09  D0A1    BNE  $DEAC
DF0B  A50F    LDA  $0F
DF0D  C9B5    CMP  #$B5
DF0F  F004    BEQ  $DF15
DF11  C9B6    CMP  #$B6
DF13  D004    BNE  $DF19
DF15  A5BE    LDA  $BE
DF17  F070    BEQ  $DF89
DF19  A901    LDA  #$01
DF1B  85BD    STA  $BD
DF1D  85B4    STA  $B4
```

```
DF1F  A900    LDA  #$00
DF21  85BE    STA  $BE
DF23  85B5    STA  $B5
DF25  85BA    STA  $BA
DF27  4C7EDF  JMP  $DF7E
DF2A  C928    CMP  #$28
DF2C  D017    BNE  $DF45
;算式中有"("的处理
DF2E  A5BA    LDA  $BA
DF30  D057    BNE  $DF89
DF32  A5BE    LDA  $BE
DF34  D053    BNE  $DF89
DF36  A5BC    LDA  $BC
DF38  D04F    BNE  $DF89
DF3A  85BD    STA  $BD
DF3C  85BE    STA  $BE
DF3E  A901    LDA  #$01
DF40  85BC    STA  $BC
DF42  4C7EDF  JMP  $DF7E
DF45  C929    CMP  #$29
DF47  D021    BNE  $DF6A
;算式中有")"的处理;
DF49  A5BC    LDA  $BC
DF4B  F03C    BEQ  $DF89
DF4D  A5BD    LDA  $BD
DF4F  D038    BNE  $DF89
DF51  A5BE    LDA  $BE
DF53  F034    BEQ  $DF89
DF55  E6B2    INC  $B2
DF57  A5B2    LDA  $B2
DF59  C90B    CMP  #$0B
DF5B  102C    BPL  $DF89
DF5D  A900    LDA  #$00
DF5F  85BC    STA  $BC
DF61  85BD    STA  $BD
DF63  A901    LDA  #$01
DF65  85BA    STA  $BA
DF67  4C7EDF  JMP  $DF7E
DF6A  C93D    CMP  #$3D
DF6C  D01B    BNE  $DF89
;算式中有"="的处理
DF6E  A5B4    LDA  $B4
DF70  F017    BEQ  $DF89
DF72  A5BC    LDA  $BC
```

```
DF74   05BD     ORA   $BD
DF76   F003     BEQ   $DF7B
DF78   4C89DF   JMP   $DF89
DF7B   A901     LDA   #$01
DF7D   60       RTS
DF7E   E8       INX
DF7F   E0D7     CPX   #$D7
DF81   9003     BCC   $DF86
DF83   4C89DF   JMP   $DF89
DF86   4CA3DE   JMP   $DEA3
       送显出错提示
DF89   A901     LDA   #$01
DF8B   2001DE   JSR   $DE01
DF8E   60       RTS
       清算式存储库
DF8F   A900     LDA   #$00
DF91   A000     LDY   #$00
DF93   990076   STA   $7600,Y
DF96   88       DEY
DF97   D0FA     BNE   $DF93
DF99   60       RTS
       运算板内容入库保存处理
DF9A   A2D8     LDX   #$D8
DF9C   BD3C06   LDA   $063C,X
DF9F   9DFF70   STA   $70FF,X
DFA2   CA       DEX
DFA3   D0F7     BNE   $DF9C
DFA5   60       RTS
       运算板的算式运算处理
DFA6   A900     LDA   #$00
DFA8   85C3     STA   $C3
DFAA   85DB     STA   $DB
DFAC   85DC     STA   $DC
DFAE   85B1     STA   $B1
DFB0   A901     LDA   #$01
DFB2   85C0     STA   $C0
DFB4   A9FF     LDA   #$FF
DFB6   85C1     STA   $C1
DFB8   A928     LDA   #$28
DFBA   8500     STA   $00
DFBC   A9FF     LDA   #$FF
DFBE   8501     STA   $01
;括号运算处理
DFC0   20F8E1   JSR   $E1F8
DFC3   D007     BNE   $DFCC
;乘、除运算处理
DFC5   20D3DF   JSR   $DFD3
;加、减运算处理
DFC8   20D3DF   JSR   $DFD3
DFCB   60       RTS
DFCC   20ECDF   JSR   $DFEC
DFCF   4CB0DF   JMP   $DFB0
DFD2   60       RTS
       乘、除运算处理子程序
DFD3   A9B5     LDA   #$B5
DFD5   8500     STA   $00
DFD7   A9B6     LDA   #$B6
DFD9   8501     STA   $01
DFDB   20F8E1   JSR   $E1F8
DFDE   F008     BEQ   $DFE8
DFE0   20BEE0   JSR   $E0BE
DFE3   A5B1     LDA   $B1
DFE5   D0EC     BNE   $DFD3
DFE7   60       RTS
DFE8   2066E0   JSR   $E066
DFEB   60       RTS
       括号运算处理子程序
DFEC   A980     LDA   #$80
DFEE   85C5     STA   $C5
DFF0   A5BF     LDA   $BF
DFF2   18       CLC
DFF3   6901     ADC   #$01
DFF5   85C0     STA   $C0
DFF7   A929     LDA   #$29
DFF9   8500     STA   $00
DFFB   A9FF     LDA   #$FF
DFFD   8501     STA   $01
DFFF   20FCE1   JSR   $E1FC
E002   A5BF     LDA   $BF
E004   85C1     STA   $C1
E006   A5C0     LDA   $C0
E008   38       SEC
E009   E901     SBC   #$01
E00B   AA       TAX
E00C   A941     LDA   #$41
E00E   9DFF75   STA   $75FF,X
E011   A6C1     LDX   $C1
E013   A942     LDA   #$42
```

```
E015    9DFF75    STA    $75FF,X
E018    20D3DF    JSR    $DFD3
E01B    20D3DF    JSR    $DFD3
E01E    A5C5      LDA    $C5
E020    297F      AND    #$7F
E022    A8        TAY
E023    A200      LDX    #$00
E025    BD3004    LDA    $0430,X
E028    998C03    STA    $038C,Y
E02B    E8        INX
E02C    C8        INY
E02D    E009      CPX    #$09
E02F    D0F4      BNE    $E025
E031    AD3A04    LDA    $043A
E034    998C03    STA    $038C,Y
E037    A6C0      LDX    $C0
E039    A5C5      LDA    $C5
E03B    9DFF75    STA    $75FF,X
E03E    A5C5      LDA    $C5
E040    18        CLC
E041    690A      ADC    #$0A
E043    85C5      STA    $C5
E045    A5C1      LDA    $C1
E047    18        CLC
E048    6901      ADC    #$01
E04A    85BF      STA    $BF
E04C    20D2E3    JSR    $E3D2
E04F    A928      LDA    #$28
E051    8500      STA    $00
E053    A9FF      LDA    #$FF
E055    8501      STA    $01
E057    20FCE1    JSR    $E1FC
E05A    48        PHA
E05B    A900      LDA    #$00
E05D    85C3      STA    $C3
E05F    68        PLA
E060    F003      BEQ    $E065
E062    4CF0DF    JMP    $DFF0
E065    60        RTS
```

CAL·板中的加、减运算子程序

```
E066    A92B      LDA    #$2B
E068    8500      STA    $00
E06A    A92D      LDA    #$2D
E06C    8501      STA    $01
E06E    20F8E1    JSR    $E1F8
E071    D001      BNE    $E074
E073    60        RTS
E074    A6BF      LDX    $BF
E076    A920      LDA    #$20
E078    9DFF75    STA    $75FF,X
E07B    A5C3      LDA    $C3
E07D    D006      BNE    $E085
;取第一数存入$400～$408;
第二数存入$410～$418
E07F    208BE3    JSR    $E38B
E082    4C88E0    JMP    $E088
E085    209AE3    JSR    $E39A
E088    A5C4      LDA    $C4
E08A    C92B      CMP    #$2B
E08C    D006      BNE    $E094
;进行加法运算
E08E    20E9E6    JSR    $E6E9
E091    4C97E0    JMP    $E097
;进行减法运算
E094    204BE7    JSR    $E74B
;加、减运算结束后的处理
E097    A209      LDX    #$09
E099    BD2F04    LDA    $042F,X
E09C    9D1F04    STA    $041F,X
E09F    CA        DEX
E0A0    D0F7      BNE    $E099
E0A2    AD3A04    LDA    $043A
E0A5    8D2A04    STA    $042A
E0A8    A901      LDA    #$01
E0AA    85C3      STA    $C3
E0AC    A5C2      LDA    $C2
E0AE    85BF      STA    $BF
E0B0    A92B      LDA    #$2B
E0B2    8500      STA    $00
E0B4    A92D      LDA    #$2D
E0B6    8501      STA    $01
E0B8    20FCE1    JSR    $E1FC
E0BB    D0B7      BNE    $E074
E0BD    60        RTS
```

CAL·板中的乘、除运算处理

```
E0BE    A900      LDA    #$00
E0C0    85B1      STA    $B1
E0C2    208BE3    JSR    $E38B
```

```
E0C5    A5C4      LDA    $C4
E0C7    C9B5      CMP    #$B5
E0C9    D006      BNE    $E0D1
;进行乘法运算
E0CB    20A3E7    JSR    $E7A3
E0CE    4CD4E0    JMP    $E0D4
;进行除法运算
E0D1    2086E8    JSR    $E886
;乘、除运算结束后的处理
E0D4    A5C2      LDA    $C2
E0D6    85BF      STA    $BF
E0D8    A6BF      LDX    $BF
E0DA    A920      LDA    #$20
E0DC    9DFF75    STA    $75FF,X
E0DF    E6BF      INC    $BF
E0E1    A6BF      LDX    $BF
;检查后面是否还有算符
E0E3    2054E2    JSR    $E254
E0E6    F017      BEQ    $E0FF
;若还有算符则把运算结果送入$400～$408
E0E8    A009      LDY    #$09
E0EA    B92F04    LDA    $042F,Y
E0ED    99FF03    STA    $03FF,Y
E0F0    88        DEY
E0F1    D0F7      BNE    $E0EA
E0F3    AD3A04    LDA    $043A
E0F6    8D0A04    STA    $040A
;取下一个数据存入$410～$418
E0F9    20F5E2    JSR    $E2F5
E0FC    4CC5E0    JMP    $E0C5
E0FF    A5C2      LDA    $C2
E101    85BF      STA    $BF
E103    C6BF      DEC    $BF
E105    A92B      LDA    #$2B
E107    8500      STA    $00
E109    A92D      LDA    #$2D
E10B    8501      STA    $01
E10D    2029E2    JSR    $E229
E110    D018      BNE    $E12A
E112    A009      LDY    #$09
E114    B92F04    LDA    $042F,Y
;暂存运算结果于$420～$428
E117    991F04    STA    $041F,Y
E11A    88        DEY
E11B    D0F7      BNE    $E114
E11D    AD3A04    LDA    $043A
E120    8D2A04    STA    $042A
E123    A901      LDA    #$01
E125    85C3      STA    $C3
E127    85B1      STA    $B1
E129    60        RTS
E12A    A5BF      LDA    $BF
E12C    85DC      STA    $DC
E12E    2084E1    JSR    $E184
E131    A5C2      LDA    $C2
E133    85BF      STA    $BF
E135    A6BF      LDX    $BF
E137    A920      LDA    #$20
E139    9DFF75    STA    $75FF,X
E13C    A5DB      LDA    $DB
E13E    C5DC      CMP    $DC
E140    D006      BNE    $E148
E142    20ADE3    JSR    $E3AD
E145    4C4BE1    JMP    $E14B
E148    20BFE3    JSR    $E3BF
E14B    A6BF      LDX    $BF
E14D    BDFF75    LDA    $75FF,X
E150    C92D      CMP    #$2D
E152    D017      BNE    $E16B
E154    A5C4      LDA    $C4
E156    C92D      CMP    #$2D
E158    D005      BNE    $E15F
E15A    A92B      LDA    #$2B
E15C    4C69E1    JMP    $E169
E15F    A92B      LDA    #$2B
E161    9DFF75    STA    $75FF,X
E164    20CBE1    JSR    $E1CB
E167    A92D      LDA    #$2D
E169    85C4      STA    $C4
E16B    A5C4      LDA    $C4
E16D    C92B      CMP    #$2B
E16F    D006      BNE    $E177
E171    20E9E6    JSR    $E6E9
E174    4CFFE0    JMP    $E0FF
E177    C92D      CMP    #$2D
E179    D006      BNE    $E181
E17B    204BE7    JSR    $E74B
E17E    4CFFE0    JMP    $E0FF
```

```
E181    A900      LDA    #$00
E183    60        RTS
E184    A900      LDA    #$00
E186    85DF      STA    $DF
E188    A4C0      LDY    $C0
E18A    A5C1      LDA    $C1
E18C    C9FF      CMP    #$FF
E18E    F007      BEQ    $E197
E190    C002      CPY    #$02
E192    F003      BEQ    $E197
E194    4C99E1    JMP    $E199
E197    A001      LDY    #$01
E199    B9FF75    LDA    $75FF,Y
E19C    C941      CMP    #$41
E19E    D007      BNE    $E1A7
E1A0    A901      LDA    #$01
E1A2    85DF      STA    $DF
E1A4    4CC3E1    JMP    $E1C3
E1A7    A5DF      LDA    $DF
E1A9    F00E      BEQ    $E1B9
E1AB    B9FF75    LDA    $75FF,Y
E1AE    C92A      CMP    #$2A
E1B0    D007      BNE    $E1B9
E1B2    A900      LDA    #$00
E1B4    85DF      STA    $DF
E1B6    4CC3E1    JMP    $E1C3
E1B9    B9FF75    LDA    $75FF,Y
E1BC    C920      CMP    #$20
E1BE    F003      BEQ    $E1C3
E1C0    84DB      STY    $DB
E1C2    60        RTS
E1C3    C8        INY
E1C4    C4C1      CPY    $C1
E1C6    90D1      BCC    $E199
E1C8    F0CF      BEQ    $E199
E1CA    60        RTS
```

两数据交换存储单元

```
E1CB    A009      LDY    #$09
E1CD    B9FF03    LDA    $03FF,Y
E1D0    99FFFF    STA    $FFFF,Y
E1D3    B90F04    LDA    $040F,Y
E1D6    99FF03    STA    $03FF,Y
E1D9    88        DEY
E1DA    D0F1      BNE    $E1CD
E1DC    AD0A04    LDA    $040A
E1DF    850F      STA    $0F
E1E1    AD1A04    LDA    $041A
E1E4    8D0A04    STA    $040A
E1E7    A009      LDY    #$09
E1E9    B9FFFF    LDA    $FFFF,Y
E1EC    990F04    STA    $040F,Y
E1EF    88        DEY
E1F0    D0F7      BNE    $E1E9
E1F2    A50F      LDA    $0F
E1F4    8D1A04    STA    $041A
E1F7    60        RTS
```

CAL·板运算符增量检索子程序

```
E1F8    A6C0      LDX    $C0
E1FA    86BF      STX    $BF
E1FC    A900      LDA    #$00
E1FE    850F      STA    $0F
E200    E4C1      CPX    $C1
E202    F022      BEQ    $E226
E204    BDFF75    LDA    $75FF,X
E207    C93D      CMP    #$3D
E209    F01B      BEQ    $E226
E20B    A500      LDA    $00
E20D    DDFF75    CMP    $75FF,X
E210    F00E      BEQ    $E220
E212    A501      LDA    $01
E214    DDFF75    CMP    $75FF,X
E217    F007      BEQ    $E220
E219    E6BF      INC    $BF
E21B    A6BF      LDX    $BF
E21D    4C00E2    JMP    $E200
E220    85C4      STA    $C4
E222    E60F      INC    $0F
E224    86C2      STX    $C2
E226    A50F      LDA    $0F
E228    60        RTS
```

CAL·板运算符减量检索子程序

```
E229    A900      LDA    #$00
E22B    850F      STA    $0F
E22D    A6BF      LDX    $BF
E22F    A5C0      LDA    $C0
E231    38        SEC
E232    E901      SBC    #$01
E234    C5BF      CMP    $BF
```

```
E236    F019     BEQ   $E251
E238    A500     LDA   $00
E23A    DDFF75   CMP   $75FF,X
E23D    F00C     BEQ   $E24B
E23F    A501     LDA   $01
E241    DDFF75   CMP   $75FF,X
E244    F005     BEQ   $E24B
E246    C6BF     DEC   $BF
E248    4C2DE2   JMP   $E22D
E24B    85C4     STA   $C4
E24D    E60F     INC   $0F
E24F    86C2     STX   $C2
E251    A50F     LDA   $0F
E253    60       RTS
```

运算符查找处理

```
E254    A900     LDA   #$00
E256    850F     STA   $0F
E258    A6BF     LDX   $BF
E25A    E4C1     CPX   $C1
E25C    F026     BEQ   $E284
E25E    BDFF75   LDA   $75FF,X
E261    C920     CMP   #$20
E263    F014     BEQ   $E279
;"="
E265    C93D     CMP   #$3D
E267    F01B     BEQ   $E284
;"+"
E269    C92B     CMP   #$2B
E26B    F017     BEQ   $E284
;"—"
E26D    C92D     CMP   #$2D
E26F    F013     BEQ   $E284
;"×"
E271    C9B5     CMP   #$B5
E273    F009     BEQ   $E27E
;"÷"
E275    C9B6     CMP   #$B6
E277    F005     BEQ   $E27E
E279    E6BF     INC   $BF
E27B    4C58E2   JMP   $E258
E27E    85C4     STA   $C4
E280    E60F     INC   $0F
E282    86C2     STX   $C2
E284    A50F     LDA   $0F
E286    60       RTS
```

括号中算式的运算处理

```
E287    A20B     LDX   #$0B
E289    A920     LDA   #$20
E28B    9DFF03   STA   $03FF,X
E28E    CA       DEX
E28F    D0F8     BNE   $E289
E291    A009     LDY   #$09
E293    C6BF     DEC   $BF
E295    A6BF     LDX   $BF
E297    A5C0     LDA   $C0
E299    38       SEC
E29A    E901     SBC   #$01
E29C    C5BF     CMP   $BF
E29E    F054     BEQ   $E2F4
E2A0    BDFF75   LDA   $75FF,X
E2A3    C920     CMP   #$20
E2A5    F0EC     BEQ   $E293
E2A7    C92B     CMP   #$2B
E2A9    F049     BEQ   $E2F4
E2AB    C92D     CMP   #$2D
E2AD    F045     BEQ   $E2F4
E2AF    C942     CMP   #$42
E2B1    F00F     BEQ   $E2C2
E2B3    BDFF75   LDA   $75FF,X
E2B6    99FF03   STA   $03FF,Y
E2B9    A920     LDA   #$20
E2BB    9DFF75   STA   $75FF,X
E2BE    88       DEY
E2BF    4C93E2   JMP   $E293
E2C2    A920     LDA   #$20
E2C4    9DFF75   STA   $75FF,X
E2C7    C6BF     DEC   $BF
E2C9    A6BF     LDX   $BF
E2CB    BD0076   LDA   $7600,X
E2CE    C941     CMP   #$41
E2D0    D0F5     BNE   $E2C7
E2D2    BD0176   LDA   $7601,X
E2D5    297F     AND   #$7F
E2D7    A8       TAY
E2D8    A920     LDA   #$20
E2DA    9D0076   STA   $7600,X
E2DD    9D0176   STA   $7601,X
E2E0    A200     LDX   #$00
```

```
E2E2   B98C03   LDA   $038C,Y
E2E5   9D0004   STA   $0400,X
E2E8   C8       INY
E2E9   E8       INX
E2EA   E009     CPX   #$09
E2EC   D0F4     BNE   $E2E2
E2EE   B98C03   LDA   $038C,Y
E2F1   8D0A04   STA   $040A
E2F4   60       RTS
;把运算数据存入四页的处理
E2F5   A20B     LDX   #$0B
E2F7   A920     LDA   #$20
E2F9   9D0F04   STA   $040F,X
E2FC   CA       DEX
E2FD   D0F8     BNE   $E2F7
E2FF   A001     LDY   #$01
E301   E6BF     INC   $BF
E303   A6BF     LDX   $BF
E305   E4C1     CPX   $C1
E307   F062     BEQ   $E36B
E309   BDFF75   LDA   $75FF,X
E30C   C920     CMP   #$20
E30E   F0F1     BEQ   $E301
E310   C93D     CMP   #$3D
E312   F057     BEQ   $E36B
E314   C92B     CMP   #$2B
E316   F053     BEQ   $E36B
E318   C92D     CMP   #$2D
E31A   F04F     BEQ   $E36B
E31C   C9B5     CMP   #$B5
E31E   F04B     BEQ   $E36B
E320   C9B6     CMP   #$B6
E322   F047     BEQ   $E36B
E324   C941     CMP   #$41
E326   F00F     BEQ   $E337
E328   BDFF75   LDA   $75FF,X
E32B   990F04   STA   $040F,Y
E32E   A920     LDA   #$20
E330   9DFF75   STA   $75FF,X
E333   C8       INY
E334   4C01E3   JMP   $E301
E337   A920     LDA   #$20
E339   9DFF75   STA   $75FF,X
E33C   BD0076   LDA   $7600,X
E33F   297F     AND   #$7F
E341   A8       TAY
E342   A920     LDA   #$20
E344   9D0076   STA   $7600,X
E347   A200     LDX   #$00
E349   B98C03   LDA   $038C,Y
E34C   9D1004   STA   $0410,X
E34F   C8       INY
E350   E8       INX
E351   E009     CPX   #$09
E353   D0F4     BNE   $E349
E355   B98C03   LDA   $038C,Y
E358   8D1A04   STA   $041A
E35B   E6BF     INC   $BF
E35D   A6BF     LDX   $BF
E35F   BDFF75   LDA   $75FF,X
E362   C942     CMP   #$42
E364   D0F5     BNE   $E35B
E366   A920     LDA   #$20
E368   9DFF75   STA   $75FF,X
E36B   A209     LDX   #$09
E36D   A009     LDY   #$09
E36F   B90F04   LDA   $040F,Y
E372   C920     CMP   #$20
E374   D004     BNE   $E37A
E376   88       DEY
E377   D0F6     BNE   $E36F
E379   60       RTS
E37A   C009     CPY   #$09
E37C   F00C     BEQ   $E38A
E37E   9D0F04   STA   $040F,X
E381   A920     LDA   #$20
E383   990F04   STA   $040F,Y
E386   CA       DEX
E387   4C76E3   JMP   $E376
E38A   60       RTS
        把两个运算数据送入四页
E38B   A5C2     LDA   $C2
E38D   85BF     STA   $BF
;第一运算数存入$400～$408
E38F   2087E2   JSR   $E287
E392   A5C2     LDA   $C2
E394   85BF     STA   $BF
;第二运算数存入$410～$418
```

```
E396    20F5E2    JSR    $E2F5
E399    60        RTS
    乘、除运算结束后进行加、减运算前的处理
E39A    A20B      LDX    #$0B
E39C    BD1F04    LDA    $041F,X
E39F    9DFF03    STA    $03FF,X
E3A2    CA        DEX
E3A3    D0F7      BNE    $E39C
E3A5    A5C2      LDA    $C2
E3A7    85BF      STA    $BF
E3A9    20F5E2    JSR    $E2F5
E3AC    60        RTS
;中间运算结果的转移处理
E3AD    A20B      LDX    #$0B
E3AF    BD1F04    LDA    $041F,X
E3B2    9DFF03    STA    $03FF,X
E3B5    BD2F04    LDA    $042F,X
E3B8    9D0F04    STA    $040F,X
E3BB    CA        DEX
E3BC    D0F1      BNE    $E3AF
E3BE    60        RTS
E3BF    A20B      LDX    #$0B
E3C1    BD2F04    LDA    $042F,X
E3C4    9D0F04    STA    $040F,X
E3C7    CA        DEX
E3C8    D0F7      BNE    $E3C1
E3CA    A5C2      LDA    $C2
E3CC    85BF      STA    $BF
E3CE    2087E2    JSR    $E287
E3D1    60        RTS
E3D2    A20B      LDX    #$0B
E3D4    A920      LDA    #$20
E3D6    9D1F04    STA    $041F,X
E3D9    CA        DEX
E3DA    D0F8      BNE    $E3D4
E3DC    60        RTS
    MUSIC 板的音名库数据
E3DD                                                    63  43  64
                                                        c   C   d
E3E0    44  65  45  66  46  67  47  61  41  62  42  00  00  31  31  32
        D   e   E   f   F   g   G   a   A   b   B           1   1   2
E3F0    32  33  33  34  34  35  35  36  36  37  37  48  4A  4C  4E  50
        2   3   3   4   4   5   5   6   6   7   7
E400    52  52  54  56  58  5A  5C  5E  60  30  32  34  36  38  3A  3A
E410    3C  3E  40  42  44  46  48  68  70  78  80  88  90  06  0B  10
E420    15  1A  1F
    MUSIC 板工作主控程序
E423    A579      LDA    $79
E425    D02B      BNE    $E452
E427    A57D      LDA    $7D
E429    D01E      BNE    $E449
E42B    A537      LDA    $37
E42D    C90B      CMP    #$0B
;按键转$E455
E42F    F024      BEQ    $E455
E431    C90C      CMP    #$0C
;按 Home+Shift 键转$E469
E433    F034      BEQ    $E469
E435    C91B      CMP    #$1B
E437    D00D      BNE    $E446
;按 ESC 键的处理
E439    20B1F4    JSR    $F4B1
E43C    A900      LDA    #$00
E43E    858B      STA    $8B
;音乐板数据入库保存
E440    206DE4    JSR    $E46D
E443    4C9BDC    JMP    $DC9B
E446    4CF6E4    JMP    $E4F6
E449    2097D1    JSR    $D197
E44C    206DE4    JSR    $E46D
E44F    4C0FF5    JMP    $F50F
E452    4C27D2    JMP    $D227
;按 Home 键的处理
E455    A9FF      LDA    #$FF
E457    857D      STA    $7D
E459    A900      LDA    #$00
E45B    858B      STA    $8B
```

```
E45D    85A1      STA   $A1
E45F    85A3      STA   $A3
E461    85A0      STA   $A0
E463    200DEF    JSR   $EF0D
E466    4C87E6    JMP   $E687
E469    A901      LDA   #$01
E46B    D0EA      BNE   $E457
```

MUSIC 板内容入库保存子程序

;$63D～$654→$7200～$7217
$655～$66C→$7260～$7277
$66D～$684→$72C0～$72D7
$69D～$6B4→$7218～$722F
$6B5～$6CC→$7278～$728F
$6CD～$6E4→$72D8～$72EF

$6FD～$714→$7230～$7247
$715～$72C→$7290～$72A7
$72D～$744→$72F0～$7307

$75D～$774→$7248～$725F
$775～$78C→$72A8～$72BF
$78D～$7A4→$7308～$731F

```
E46D    A200      LDX   #$00
E46F    A000      LDY   #$00
E471    B93D06    LDA   $063D,Y
E474    9D0072    STA   $7200,X
E477    B95506    LDA   $0655,Y
E47A    9D6072    STA   $7260,X
E47D    B96D06    LDA   $066D,Y
E480    9DC072    STA   $72C0,X
E483    E8        INX
E484    C8        INY
E485    C018      CPY   #$18
E487    D002      BNE   $E48B
E489    A060      LDY   #$60
E48B    E030      CPX   #$30
E48D    D0E2      BNE   $E471
E48F    A000      LDY   #$00
E491    B9FD06    LDA   $06FD,Y
E494    9D0072    STA   $7200,X
E497    B91507    LDA   $0715,Y
E49A    9D6072    STA   $7260,X
E49D    B92D07    LDA   $072D,Y
E4A0    9DC072    STA   $72C0,X
E4A3    E8        INX
E4A4    C8        INY
E4A5    C018      CPY   #$18
E4A7    D002      BNE   $E4AB
E4A9    A060      LDY   #$60
E4AB    E060      CPX   #$60
E4AD    D0E2      BNE   $E491
E4AF    60        RTS
```

;把音乐板数据库的数据送回音乐板
送回是存入的逆过程(从略)

```
E4B0    2060D2    JSR   $D260
E4B3    A200      LDX   #$00
E4B5    A000      LDY   #$00
E4B7    BD0072    LDA   $7200,X
E4BA    993D06    STA   $063D,Y
E4BD    BD6072    LDA   $7260,X
E4C0    995506    STA   $0655,Y
E4C3    BDC072    LDA   $72C0,X
E4C6    996D06    STA   $066D,Y
E4C9    E8        INX
E4CA    C8        INY
E4CB    C018      CPY   #$18
E4CD    D002      BNE   $E4D1
E4CF    A060      LDY   #$60
E4D1    E030      CPX   #$30
E4D3    D0E2      BNE   $E4B7
E4D5    A000      LDY   #$00
E4D7    BD0072    LDA   $7200,X
E4DA    99FD06    STA   $06FD,Y
E4DD    BD6072    LDA   $7260,X
E4E0    991507    STA   $0715,Y
E4E3    BDC072    LDA   $72C0,X
E4E6    992D07    STA   $072D,Y
E4E9    E8        INX
E4EA    C8        INY
E4EB    C018      CPY   #$18
E4ED    D002      BNE   $E4F1
E4EF    A060      LDY   #$60
E4F1    E060      CPX   #$60
E4F3    D0E2      BNE   $E4D7
E4F5    60        RTS
E4F6    A204      LDX   #$04
E4F8    202CEF    JSR   $EF2C
E4FB    A58B      LDA   $8B
```

```
E4FD    D02C      BNE   $E52B
E4FF    A538      LDA   $38
E501    C9C8      CMP   #$C8
E503    D055      BNE   $E55A
      MUSIC 板演奏处理
E505    206DE4    JSR   $E46D
E508    E68B      INC   $8B
;定义向下羽毛笔
E50A    20E8EF    JSR   $EFE8
E50D    A900      LDA   #$00
E50F    85A1      STA   $A1
E511    A900      LDA   #$00
E513    85A0      STA   $A0
;取音符送入$344、$345、$346
E515    2096E5    JSR   $E596
E518    A900      LDA   #$00
E51A    85A0      STA   $A0
E51C    85A3      STA   $A3
E51E    85A4      STA   $A4
E520    2087E6    JSR   $E687
E523    A68C      LDX   $8C
E525    BD1DE4    LDA   $E41D,X
E528    8524      STA   $24
E52A    60        RTS
      演奏速度的调整
E52B    A5A1      LDA   $A1
E52D    D02C      BNE   $E55B
E52F    A540      LDA   $40
E531    C91C      CMP   #$1C
E533    D00A      BNE   $E53F
;减慢演奏速度
E535    A68C      LDX   $8C
E537    E005      CPX   #$05
E539    F046      BEQ   $E581
E53B    E8        INX
E53C    4C48E5    JMP   $E548
E53F    C91D      CMP   #$1D
E541    D00D      BNE   $E550
;加快演奏速度
E543    A68C      LDX   $8C
E545    F03A      BEQ   $E581
E547    CA        DEX
E548    868C      STX   $8C
E54A    20D2E6    JSR   $E6D2
E54D    4C81E5    JMP   $E581
E550    C9C8      CMP   #$C8
E552    F0BD      BEQ   $E511
E554    C920      CMP   #$20
E556    D029      BNE   $E581
E558    E6A1      INC   $A1
E55A    60        RTS
E55B    A540      LDA   $40
E55D    C9C8      CMP   #$C8
E55F    D005      BNE   $E566
E561    A900      LDA   #$00
E563    85A1      STA   $A1
E565    60        RTS
;单个演奏音符的处理
E566    C91C      CMP   #$1C
E568    D00C      BNE   $E576
E56A    A5A0      LDA   $A0
E56C    C95F      CMP   #$5F
E56E    1010      BPL   $E580
E570    2041E6    JSR   $E641
E573    4C96E5    JMP   $E596
E576    C91D      CMP   #$1D
E578    D006      BNE   $E580
E57A    2067E6    JSR   $E667
E57D    4C96E5    JMP   $E596
E580    60        RTS
E581    A524      LDA   $24
E583    D0D5      BNE   $E55A
E585    A68C      LDX   $8C
E587    BD1DE4    LDA   $E41D,X
E58A    8524      STA   $24
E58C    2041E6    JSR   $E641
E58F    A5A4      LDA   $A4
E591    F003      BEQ   $E596
E593    4C11E5    JMP   $E511
;演奏过程控制
E596    A4A0      LDY   $A0
E598    C060      CPY   #$60
E59A    101E      BPL   $E5BA
E59C    B90072    LDA   $7200,Y
E59F    C924      CMP   #$24
E5A1    F01E      BEQ   $E5C1
E5A3    20CFE5    JSR   $E5CF
E5A6    B96072    LDA   $7260,Y
```

```
E5A9   C924     CMP   #$24
E5AB   F014     BEQ   $E5C1
E5AD   20F4E5   JSR   $E5F4
E5B0   B9C072   LDA   $72C0,Y
E5B3   C924     CMP   #$24
E5B5   F00A     BEQ   $E5C1
E5B7   4C19E6   JMP   $E619
E5BA   A900     LDA   #$00
E5BC   858B     STA   $8B
E5BE   4C18E5   JMP   $E518
E5C1   C000     CPY   #$00
E5C3   F005     BEQ   $E5CA
E5C5   A901     LDA   #$01
E5C7   85A4     STA   $A4
E5C9   60       RTS
E5CA   A901     LDA   #$01
E5CC   85A3     STA   $A3
E5CE   60       RTS
```

第一声部的处理

```
E5CF   A200     LDX   #$00
E5D1   DDDDE3   CMP   $E3DD,X
E5D4   F008     BEQ   $E5DE
E5D6   E8       INX
E5D7   E01E     CPX   #$1E
E5D9   30F6     BMI   $E5D1
E5DB   A900     LDA   #$00
E5DD   60       RTS
E5DE   E010     CPX   #$10
E5E0   1007     BPL   $E5E9
E5E2   BDFBE3   LDA   $E3FB,X
E5E5   8D4403   STA   $0344
E5E8   60       RTS
E5E9   8A       TXA
E5EA   290F     AND   #$0F
E5EC   AA       TAX
E5ED   BD09E4   LDA   $E409,X
E5F0   8D4403   STA   $0344
E5F3   60       RTS
```

第二声部的处理

```
E5F4   A200     LDX   #$00
E5F6   DDDDE3   CMP   $E3DD,X
E5F9   F008     BEQ   $E603
E5FB   E8       INX
E5FC   E01E     CPX   #$1E
E5FE   30F6     BMI   $E5F6
E600   A900     LDA   #$00
E602   60       RTS
E603   E010     CPX   #$10
E605   1007     BPL   $E60E
E607   BD09E4   LDA   $E409,X
E60A   8D4503   STA   $0345
E60D   60       RTS
E60E   8A       TXA
E60F   290F     AND   #$0F
E611   AA       TAX
E612   BDFBE3   LDA   $E3FB,X
E615   8D4503   STA   $0345
E618   60       RTS
```

第三声部的处理

```
E619   A200     LDX   #$00
E61B   DDDDE3   CMP   $E3DD,X
E61E   F008     BEQ   $E628
E620   E8       INX
E621   E01E     CPX   #$1E
E623   30F6     BMI   $E61B
E625   A900     LDA   #$00
E627   60       RTS
E628   E010     CPX   #$10
E62A   1007     BPL   $E633
E62C   BD09E4   LDA   $E409,X
E62F   8D4603   STA   $0346
E632   60       RTS
E633   8A       TXA
E634   290F     AND   #$0F
E636   AA       TAX
E637   BD09E4   LDA   $E409,X
E63A   38       SEC
E63B   E918     SBC   #$18
E63D   8D4603   STA   $0346
E640   60       RTS
```

演奏完第一个字符后的处理

```
E641   A5A3     LDA   $A3
E643   D021     BNE   $E666
E645   A5A0     LDA   $A0
E647   C960     CMP   #$60
E649   101B     BPL   $E666
E64B   E6A0     INC   $A0
E64D   A53E     LDA   $3E
```

```
E64F    18        CLC
E650    6908      ADC    #$08
E652    853E      STA    $3E
E654    C9E0      CMP    #$E0
E656    D00B      BNE    $E663
E658    A920      LDA    #$20
E65A    853E      STA    $3E
E65C    A53F      LDA    $3F
E65E    18        CLC
E65F    6920      ADC    #$20
E661    853F      STA    $3F
E663    208FE6    JSR    $E68F
E666    60        RTS
```

倒演奏的处理

```
E667    A5A0      LDA    $A0
E669    F01B      BEQ    $E686
E66B    C6A0      DEC    $A0
E66D    A53E      LDA    $3E
E66F    38        SEC
E670    E908      SBC    #$08
E672    853E      STA    $3E
E674    C918      CMP    #$18
E676    D00B      BNE    $E683
E678    A9D8      LDA    #$D8
E67A    853E      STA    $3E
E67C    A53F      LDA    $3F
E67E    38        SEC
E67F    E920      SBC    #$20
E681    853F      STA    $3F
E683    208FE6    JSR    $E68F
E686    60        RTS
```

定义三音框卡通的处理

```
E687    A920      LDA    #$20
E689    853E      STA    $3E
E68B    A937      LDA    #$37
E68D    853F      STA    $3F
E68F    A53E      LDA    $3E
E691    8500      STA    $00
E693    C600      DEC    $00
E695    A53F      LDA    $3F
E697    8501      STA    $01
E699    A9D9      LDA    #$D9
E69B    8502      STA    $02
E69D    A913      LDA    #$13
E69F    8503      STA    $03
E6A1    A900      LDA    #$00
E6A3    8504      STA    $04
E6A5    20BEEC    JSR    $ECBE
E6A8    A9D9      LDA    #$D9
E6AA    8D0502    STA    $0205
E6AD    8D0902    STA    $0209
E6B0    A53E      LDA    $3E
E6B2    8500      STA    $00
E6B4    A53F      LDA    $3F
E6B6    8501      STA    $01
E6B8    C601      DEC    $01
E6BA    A9DA      LDA    #$DA
E6BC    8502      STA    $02
E6BE    A913      LDA    #$13
E6C0    8503      STA    $03
E6C2    A90C      LDA    #$0C
E6C4    8504      STA    $04
E6C6    20BEEC    JSR    $ECBE
E6C9    A9DA      LDA    #$DA
E6CB    8D1102    STA    $0211
E6CE    8D1502    STA    $0215
E6D1    60        RTS
E6D2    BD17E4    LDA    $E417,X
E6D5    8D2302    STA    $0223
E6D8    A901      LDA    #$01
E6DA    8D2202    STA    $0222
E6DD    A9DB      LDA    #$DB
E6DF    8D2102    STA    $0221
E6E2    A9B7      LDA    #$B7
E6E4    8D2002    STA    $0220
E6E7    60        RTS
E6E8    FF        ???
```

CAL·板工作程序

```
E6E9    A900      LDA    #$00
E6EB    8509      STA    $09
E6ED    20F1E9    JSR    $E9F1
E6F0    2052EA    JSR    $EA52
E6F3    20FDEB    JSR    $EBFD
E6F6    C9FF      CMP    #$FF
E6F8    D003      BNE    $E6FD
E6FA    4C5FE7    JMP    $E75F
```

加法运算处理

```
E6FD    18        CLC
```

```
E6FE    A208      LDX    #$08
E700    BD0004    LDA    $0400,X
E703    A409      LDY    $09
E705    F002      BEQ    $E709
E707    B597      LDA    $97,X
E709    7D1004    ADC    $0410,X
E70C    9D3004    STA    $0430,X
E70F    9597      STA    $97,X
E711    69F6      ADC    #$F6
E713    9005      BCC    $E71A
E715    9D3004    STA    $0430,X
E718    9597      STA    $97,X
E71A    BD3004    LDA    $0430,X
E71D    0930      ORA    #$30
E71F    9D3004    STA    $0430,X
E722    CA        DEX
E723    10DB      BPL    $E700
E725    9012      BCC    $E739
;运算结果有进位的处理
E727    A208      LDX    #$08
E729    F68E      INC    $8E,X
E72B    B58E      LDA    $8E,X
E72D    C90A      CMP    #$0A
E72F    3008      BMI    $E739
E731    A900      LDA    #$00
E733    958E      STA    $8E,X
E735    CA        DEX
E736    10F1      BPL    $E729
E738    60        RTS
E739    A509      LDA    $09
E73B    F001      BEQ    $E73E
E73D    60        RTS
E73E    AD0904    LDA    $0409
E741    8D3904    STA    $0439
E744    20D3E9    JSR    $E9D3
E747    208BEA    JSR    $EA8B
E74A    60        RTS
```

减法运算处理

```
E74B    A900      LDA    #$00
E74D    8509      STA    $09
E74F    20F1E9    JSR    $E9F1
E752    2052EA    JSR    $EA52
E755    2013EC    JSR    $EC13
E758    C9FF      CMP    #$FF
E75A    D003      BNE    $E75F
E75C    4CFDE6    JMP    $E6FD
;比较两运算数的大小
E75F    20E6EB    JSR    $EBE6
E762    F00C      BEQ    $E770
;若减数的绝对值大则交换位置
E764    20C6EB    JSR    $EBC6
E767    AD1A04    LDA    $041A
E76A    8D3A04    STA    $043A
E76D    4C76E7    JMP    $E776
E770    AD0A04    LDA    $040A
E773    8D3A04    STA    $043A
;按位减处理
E776    A208      LDX    #$08
E778    BD0004    LDA    $0400,X
E77B    DD1004    CMP    $0410,X
E77E    1006      BPL    $E786
E780    DEFF03    DEC    $03FF,X
E783    18        CLC
E784    690A      ADC    #$0A
E786    38        SEC
E787    FD1004    SBC    $0410,X
E78A    9D0004    STA    $0400,X
E78D    0930      ORA    #$30
E78F    9D3004    STA    $0430,X
E792    CA        DEX
E793    10E3      BPL    $E778
E795    A509      LDA    $09
E797    D009      BNE    $E7A2
E799    AD0904    LDA    $0409
E79C    8D3904    STA    $0439
;对运算结果加小数点
E79F    208BEA    JSR    $EA8B
E7A2    60        RTS
E7A3    20F1E9    JSR    $E9F1
E7A6    204CEC    JSR    $EC4C
E7A9    203CEC    JSR    $EC3C
E7AC    20DFEA    JSR    $EADF
E7AF    F005      BEQ    $E7B6
E7B1    20ECEA    JSR    $EAEC
E7B4    D003      BNE    $E7B9
E7B6    4C1DEB    JMP    $EB1D
```

乘法运算处理

```
E7B9    A908      LDA    #$08
```

```
E7BB   8509     STA   $09
E7BD   850A     STA   $0A
E7BF   AD0104   LDA   $0401
E7C2   F008     BEQ   $E7CC
E7C4   20FDE6   JSR   $E6FD
E7C7   CE0104   DEC   $0401
E7CA   D0F3     BNE   $E7BF
E7CC   C60A     DEC   $0A
E7CE   F009     BEQ   $E7D9
E7D0   203FEB   JSR   $EB3F
E7D3   204AEB   JSR   $EB4A
E7D6   4CBFE7   JMP   $E7BF
;对乘积的处理
E7D9   A200     LDX   #$00
E7DB   B58E     LDA   $8E,X
E7DD   D014     BNE   $E7F3
E7DF   E60A     INC   $0A
E7E1   E8       INX
E7E2   E012     CPX   #$12
E7E4   D0F5     BNE   $E7DB
E7E6   A900     LDA   #$00
E7E8   A20A     LDX   #$0A
E7EA   9D3004   STA   $0430,X
E7ED   CA       DEX
E7EE   10FA     BPL   $E7EA
E7F0   4C81E8   JMP   $E881
E7F3   A900     LDA   #$00
E7F5   850B     STA   $0B
E7F7   A50A     LDA   $0A
E7F9   C908     CMP   #$08
E7FB   1010     BPL   $E80D
E7FD   2034EB   JSR   $EB34
E800   E60A     INC   $0A
E802   AD1904   LDA   $0419
E805   F01B     BEQ   $E822
E807   CE1904   DEC   $0419
E80A   4CF7E7   JMP   $E7F7
E80D   AD1904   LDA   $0419
E810   C907     CMP   #$07
E812   F028     BEQ   $E83C
E814   AD0904   LDA   $0409
E817   F023     BEQ   $E83C
E819   CE0904   DEC   $0409
E81C   EE1904   INC   $0419
E81F   4C0DE8   JMP   $E80D
E822   A50B     LDA   $0B
E824   F003     BEQ   $E829
E826   4C7DE8   JMP   $E87D
E829   A901     LDA   #$01
E82B   850B     STA   $0B
E82D   AD0904   LDA   $0409
E830   AC1904   LDY   $0419
E833   8D1904   STA   $0419
E836   8C0904   STY   $0409
E839   4C02E8   JMP   $E802
;小数乘积的四舍五入处理
E83C   AD0904   LDA   $0409
E83F   F008     BEQ   $E849
E841   2034EB   JSR   $EB34
E844   CE0904   DEC   $0409
E847   D0F3     BNE   $E83C
E849   AD1904   LDA   $0419
E84C   F00E     BEQ   $E85C
E84E   A596     LDA   $96
E850   0597     ORA   $97
E852   F008     BEQ   $E85C
E854   2034EB   JSR   $EB34
E857   CE1904   DEC   $0419
E85A   D0F2     BNE   $E84E
E85C   A208     LDX   #$08
E85E   B597     LDA   $97,X
E860   0930     ORA   #$30
E862   9D3004   STA   $0430,X
E865   CA       DEX
E866   10F6     BPL   $E85E
E868   AD1904   LDA   $0419
E86B   8D3904   STA   $0439
E86E   208BEA   JSR   $EA8B
E871   AD3904   LDA   $0439
E874   D00B     BNE   $E881
E876   AD3004   LDA   $0430
E879   C930     CMP   #$30
E87B   F004     BEQ   $E881
E87D   A902     LDA   #$02
E87F   85B9     STA   $B9
E881   A900     LDA   #$00
E883   8509     STA   $09
E885   60       RTS
```

除法运算处理

```
E886   20F1E9    JSR   $E9F1
E889   204CEC    JSR   $EC4C
E88C   20ECEA    JSR   $EAEC
E88F   D007      BNE   $E898
E891   A901      LDA   #$01
E893   85B9      STA   $B9
E895   4C1DEB    JMP   $EB1D
E898   20DFEA    JSR   $EADF
E89B   F0F8      BEQ   $E895
E89D   A901      LDA   #$01
E89F   8509      STA   $09
E8A1   A900      LDA   #$00
E8A3   850A      STA   $0A
E8A5   850B      STA   $0B
E8A7   850C      STA   $0C
E8A9   203CEC    JSR   $EC3C
E8AC   AD1904    LDA   $0419
E8AF   CD0904    CMP   $0409
E8B2   1007      BPL   $E8BB
E8B4   20D9EB    JSR   $EBD9
E8B7   A901      LDA   #$01
E8B9   850C      STA   $0C
E8BB   AD1904    LDA   $0419
E8BE   38        SEC
E8BF   ED0904    SBC   $0409
E8C2   8D1904    STA   $0419
E8C5   A900      LDA   #$00
E8C7   8D0904    STA   $0409
E8CA   A50C      LDA   $0C
E8CC   F007      BEQ   $E8D5
E8CE   20D9EB    JSR   $EBD9
E8D1   A900      LDA   #$00
E8D3   850C      STA   $0C
E8D5   AD1004    LDA   $0410
E8D8   D008      BNE   $E8E2
E8DA   2060EB    JSR   $EB60
E8DD   E60B      INC   $0B
E8DF   4CD5E8    JMP   $E8D5
E8E2   AD0004    LDA   $0400
E8E5   D00B      BNE   $E8F2
E8E7   204AEB    JSR   $EB4A
E8EA   E60A      INC   $0A
E8EC   A50A      LDA   $0A
E8EE   C908      CMP   #$08
E8F0   D0F0      BNE   $E8E2
E8F2   A50A      LDA   $0A
E8F4   C50B      CMP   $0B
E8F6   B004      BCS   $E8FC
E8F8   A901      LDA   #$01
E8FA   850C      STA   $0C
E8FC   A50A      LDA   $0A
E8FE   38        SEC
E8FF   E50B      SBC   $0B
E901   850A      STA   $0A
E903   A900      LDA   #$00
E905   850B      STA   $0B
;比较被除数、除数的大小
E907   20E6EB    JSR   $EBE6
E90A   F005      BEQ   $E911
E90C   2055EB    JSR   $EB55
E90F   E60A      INC   $0A
E911   A50C      LDA   $0C
E913   F00F      BEQ   $E924
E915   A50A      LDA   $0A
E917   49FF      EOR   #$FF
E919   A60B      LDX   $0B
E91B   850B      STA   $0B
E91D   E60B      INC   $0B
E91F   860A      STX   $0A
E921   4C35E9    JMP   $E935
E924   AD1904    LDA   $0419
E927   F00C      BEQ   $E935
E929   A50A      LDA   $0A
E92B   F008      BEQ   $E935
E92D   C60A      DEC   $0A
E92F   CE1904    DEC   $0419
E932   4C24E9    JMP   $E924
;以减代除运算
E935   2076E7    JSR   $E776
E938   E69F      INC   $9F
E93A   20E6EB    JSR   $EBE6
E93D   F0F6      BEQ   $E935
E93F   E60A      INC   $0A
E941   20DFEA    JSR   $EADF
E944   F01A      BEQ   $E960
E946   A50A      LDA   $0A
E948   C908      CMP   #$08
```

```
;商的位数达到八位时结束运算
E94A    F014      BEQ    $E960
E94C    203FEB    JSR    $EB3F
E94F    AD0004    LDA    $0400
E952    F006      BEQ    $E95A
E954    2055EB    JSR    $EB55
E957    4C3AE9    JMP    $E93A
E95A    204AEB    JSR    $EB4A
E95D    4C3AE9    JMP    $E93A
;达到运算精度的处理
E960    A50B      LDA    $0B
E962    F016      BEQ    $E97A
E964    A50A      LDA    $0A
E966    38        SEC
E967    E50B      SBC    $0B
E969    850A      STA    $0A
E96B    F002      BEQ    $E96F
E96D    100B      BPL    $E97A
E96F    203FEB    JSR    $EB3F
E972    E60A      INC    $0A
E974    A50A      LDA    $0A
E976    C901      CMP    #$01
E978    D0F5      BNE    $E96F
E97A    2089E9    JSR    $E989
E97D    2010EB    JSR    $EB10
E980    20CCE9    JSR    $E9CC
E983    208BEA    JSR    $EA8B
E986    8509      STA    $09
E988    60        RTS
;除运算结束后对商的整形处理
E989    AD0904    LDA    $0409
E98C    F014      BEQ    $E9A2
E98E    A50A      LDA    $0A
E990    C909      CMP    #$09
E992    D006      BNE    $E99A
E994    2034EB    JSR    $EB34
E997    4C9CE9    JMP    $E99C
E99A    E60A      INC    $0A
E99C    CE0904    DEC    $0409
E99F    4C89E9    JMP    $E989
E9A2    AD1904    LDA    $0419
E9A5    F01D      BEQ    $E9C4
E9A7    A50A      LDA    $0A
E9A9    C901      CMP    #$01
E9AB    F008      BEQ    $E9B5
E9AD    C60A      DEC    $0A
E9AF    CE1904    DEC    $0419
E9B2    4CA2E9    JMP    $E9A2
E9B5    A596      LDA    $96
E9B7    F005      BEQ    $E9BE
E9B9    A902      LDA    #$02
E9BB    85B9      STA    $B9
E9BD    60        RTS
E9BE    203FEB    JSR    $EB3F
E9C1    4CAFE9    JMP    $E9AF
E9C4    C60A      DEC    $0A
E9C6    A50A      LDA    $0A
E9C8    8D3904    STA    $0439
E9CB    60        RTS
;检查商是否有溢出
E9CC    AD3904    LDA    $0439
E9CF    C908      CMP    #$08
E9D1    F012      BEQ    $E9E5
E9D3    AD3004    LDA    $0430
E9D6    C930      CMP    #$30
E9D8    D001      BNE    $E9DB
E9DA    60        RTS
E9DB    AD3904    LDA    $0439
E9DE    D005      BNE    $E9E5
E9E0    A902      LDA    #$02
E9E2    85B9      STA    $B9
E9E4    60        RTS
E9E5    CE3904    DEC    $0439
E9E8    206BEB    JSR    $EB6B
E9EB    A930      LDA    #$30
E9ED    8D3004    STA    $0430
E9F0    60        RTS
```

还原运算数子程序

```
E9F1    A900      LDA    #$00
E9F3    8500      STA    $00
E9F5    A904      LDA    #$04
E9F7    8501      STA    $01
E9F9    A200      LDX    #$00
E9FB    2008EA    JSR    $EA08
E9FE    A910      LDA    #$10
EA00    8500      STA    $00
EA02    A904      LDA    #$04
EA04    8501      STA    $01
```

```
EA06   A201     LDX   #$01
EA08   204CEC   JSR   $EC4C
EA0B   A900     LDA   #$00
EA0D   850F     STA   $0F
EA0F   8A       TXA
EA10   48       PHA
EA11   A008     LDY   #$08
EA13   A208     LDX   #$08
EA15   B100     LDA   ($00),Y
EA17   48       PHA
EA18   C92E     CMP   #$2E
EA1A   D005     BNE   $EA21
EA1C   85B8     STA   $B8
EA1E   68       PLA
EA1F   D00C     BNE   $EA2D
EA21   A5B8     LDA   $B8
EA23   D002     BNE   $EA27
EA25   E60F     INC   $0F
EA27   68       PLA
EA28   290F     AND   #$0F
EA2A   958E     STA   $8E,X
EA2C   CA       DEX
EA2D   88       DEY
EA2E   10E5     BPL   $EA15
EA30   68       PLA
EA31   0A       ASL
EA32   0A       ASL
EA33   0A       ASL
EA34   0A       ASL
EA35   AA       TAX
EA36   A50F     LDA   $0F
EA38   C909     CMP   #$09
EA3A   3002     BMI   $EA3E
EA3C   A900     LDA   #$00
EA3E   9D0904   STA   $0409,X
EA41   A008     LDY   #$08
EA43   A208     LDX   #$08
EA45   B58E     LDA   $8E,X
EA47   9100     STA   ($00),Y
EA49   CA       DEX
EA4A   88       DEY
EA4B   10F8     BPL   $EA45
EA4D   A900     LDA   #$00
EA4F   85B8     STA   $B8
EA51   60       RTS
        小数运算的处理
EA52   A900     LDA   #$00
EA54   850E     STA   $0E
EA56   AD0904   LDA   $0409
EA59   CD1904   CMP   $0419
EA5C   F02C     BEQ   $EA8A
EA5E   3007     BMI   $EA67
EA60   A901     LDA   #$01
EA62   850E     STA   $0E
EA64   20C6EB   JSR   $EBC6
EA67   AD0104   LDA   $0401
EA6A   F009     BEQ   $EA75
EA6C   2055EB   JSR   $EB55
EA6F   CE1904   DEC   $0419
EA72   4C7BEA   JMP   $EA7B
;使两数的小数点对齐
EA75   204AEB   JSR   $EB4A
EA78   EE0904   INC   $0409
EA7B   AD0904   LDA   $0409
EA7E   CD1904   CMP   $0419
EA81   D0E4     BNE   $EA67
EA83   A50E     LDA   $0E
EA85   F003     BEQ   $EA8A
EA87   20C6EB   JSR   $EBC6
EA8A   60       RTS
        对结果添小数点的处理
EA8B   20F9EA   JSR   $EAF9
EA8E   AD3904   LDA   $0439
EA91   D001     BNE   $EA94
EA93   60       RTS
EA94   AD3804   LDA   $0438
EA97   C930     CMP   #$30
EA99   D006     BNE   $EAA1
EA9B   20D0EA   JSR   $EAD0
EA9E   D0F4     BNE   $EA94
EAA0   60       RTS
EAA1   AD3004   LDA   $0430
EAA4   C930     CMP   #$30
EAA6   F006     BEQ   $EAAE
EAA8   20D0EA   JSR   $EAD0
EAAB   D001     BNE   $EAAE
EAAD   60       RTS
EAAE   A008     LDY   #$08
```

```
EAB0  A908     LDA  #$08
EAB2  38       SEC
EAB3  ED3904   SBC  $0439
EAB6  850B     STA  $0B
EAB8  AA       TAX
EAB9  18       CLC
EABA  BD3004   LDA  $0430,X
EABD  2A       ROL
EABE  9D3004   STA  $0430,X
EAC1  CA       DEX
EAC2  10F6     BPL  $EABA
EAC4  88       DEY
EAC5  D0E9     BNE  $EAB0
EAC7  A50B     LDA  $0B
EAC9  AA       TAX
EACA  A92E     LDA  #$2E
EACC  9D3004   STA  $0430,X
EACF  60       RTS
```

;删除小数部分最后的零

```
EAD0  206BEB   JSR  $EB6B
EAD3  A930     LDA  #$30
EAD5  8D3004   STA  $0430
EAD8  CE3904   DEC  $0439
EADB  AD3904   LDA  $0439
EADE  60       RTS
EADF  A208     LDX  #$08
EAE1  BD0004   LDA  $0400,X
EAE4  D005     BNE  $EAEB
EAE6  CA       DEX
EAE7  10F8     BPL  $EAE1
EAE9  A900     LDA  #$00
EAEB  60       RTS
EAEC  A208     LDX  #$08
EAEE  BD1004   LDA  $0410,X
EAF1  D005     BNE  $EAF8
EAF3  CA       DEX
EAF4  10F8     BPL  $EAEE
EAF6  A900     LDA  #$00
EAF8  60       RTS
```

运算结果的整形和判断处理

```
EAF9  A208     LDX  #$08
EAFB  BD3004   LDA  $0430,X
EAFE  C930     CMP  #$30
EB00  D00D     BNE  $EB0F
EB02  CA       DEX
EB03  10F6     BPL  $EAFB
EB05  A920     LDA  #$20
EB07  8D3A04   STA  $043A
EB0A  A900     LDA  #$00
EB0C  8D3904   STA  $0439
EB0F  60       RTS
EB10  A208     LDX  #$08
EB12  B597     LDA  $97,X
EB14  0930     ORA  #$30
EB16  9D3004   STA  $0430,X
EB19  CA       DEX
EB1A  10F6     BPL  $EB12
EB1C  60       RTS
```

;乘运算中有一数为零时令积为零

```
EB1D  A208     LDX  #$08
EB1F  A930     LDA  #$30
EB21  9D3004   STA  $0430,X
EB24  CA       DEX
EB25  10FA     BPL  $EB21
EB27  A920     LDA  #$20
EB29  8D3A04   STA  $043A
EB2C  A900     LDA  #$00
EB2E  8D3904   STA  $0439
EB31  8509     STA  $09
EB33  60       RTS
```

运算系列子程序

;后移位处理

```
EB34  A98E     LDA  #$8E
EB36  8500     STA  $00
EB38  A900     LDA  #$00
EB3A  8501     STA  $01
EB3C  4CAFEB   JMP  $EBAF
```

;前移位处理

```
EB3F  A98E     LDA  #$8E
EB41  8500     STA  $00
EB43  A900     LDA  #$00
EB45  8501     STA  $01
EB47  4C87EB   JMP  $EB87
```

;前移位处理

```
EB4A  A900     LDA  #$00
EB4C  8500     STA  $00
EB4E  A904     LDA  #$04
EB50  8501     STA  $01
```

```
EB52   4C76EB   JMP   $EB76
;后移位处理
EB55   A910     LDA   #$10
EB57   8500     STA   $00
EB59   A904     LDA   #$04
EB5B   8501     STA   $01
EB5D   4C98EB   JMP   $EB98
;前移位处理
EB60   A910     LDA   #$10
EB62   8500     STA   $00
EB64   A904     LDA   #$04
EB66   8501     STA   $01
EB68   4C76EB   JMP   $EB76
;后移位处理
EB6B   A930     LDA   #$30
EB6D   8500     STA   $00
EB6F   A904     LDA   #$04
EB71   8501     STA   $01
EB73   4C98EB   JMP   $EB98
;前移位处理
EB76   A208     LDX   #$08
EB78   18       CLC
EB79   A008     LDY   #$08
EB7B   B100     LDA   ($00),Y
EB7D   2A       ROL
EB7E   9100     STA   ($00),Y
EB80   88       DEY
EB81   10F8     BPL   $EB7B
EB83   CA       DEX
EB84   D0F2     BNE   $EB78
EB86   60       RTS
;前移位处理
EB87   A208     LDX   #$08
EB89   18       CLC
EB8A   A011     LDY   #$11
EB8C   B100     LDA   ($00),Y
EB8E   2A       ROL
EB8F   9100     STA   ($00),Y
EB91   88       DEY
EB92   10F8     BPL   $EB8C
EB94   CA       DEX
EB95   D0F2     BNE   $EB89
EB97   60       RTS
;后移位处理
EB98   A208     LDX   #$08
EB9A   A908     LDA   #$08
EB9C   8508     STA   $08
EB9E   18       CLC
EB9F   A000     LDY   #$00
EBA1   B100     LDA   ($00),Y
EBA3   6A       ROR
EBA4   9100     STA   ($00),Y
EBA6   C8       INY
EBA7   C608     DEC   $08
EBA9   10F6     BPL   $EBA1
EBAB   CA       DEX
EBAC   D0EC     BNE   $EB9A
EBAE   60       RTS
;后移位处理
EBAF   A208     LDX   #$08
EBB1   A911     LDA   #$11
EBB3   8508     STA   $08
EBB5   18       CLC
EBB6   A000     LDY   #$00
EBB8   B100     LDA   ($00),Y
EBBA   6A       ROR
EBBB   9100     STA   ($00),Y
EBBD   C8       INY
EBBE   C608     DEC   $08
EBC0   10F6     BPL   $EBB8
EBC2   CA       DEX
EBC3   D0EC     BNE   $EBB1
EBC5   60       RTS
;数据交换子程序
EBC6   A208     LDX   #$08
EBC8   BD0004   LDA   $0400,X
EBCB   A8       TAY
EBCC   BD1004   LDA   $0410,X
EBCF   9D0004   STA   $0400,X
EBD2   98       TYA
EBD3   9D1004   STA   $0410,X
EBD6   CA       DEX
EBD7   10EF     BPL   $EBC8
EBD9   AD0904   LDA   $0409
EBDC   AE1904   LDX   $0419
EBDF   8E0904   STX   $0409
EBE2   8D1904   STA   $0419
EBE5   60       RTS
```

```
;比较大小子程序
EBE6    A200      LDX   #$00
EBE8    BD0004    LDA   $0400,X
EBEB    DD1004    CMP   $0410,X
EBEE    F005      BEQ   $EBF5
EBF0    1008      BPL   $EBFA
EBF2    A901      LDA   #$01
EBF4    60        RTS
EBF5    E8        INX
EBF6    E009      CPX   #$09
EBF8    30EE      BMI   $EBE8
EBFA    A900      LDA   #$00
EBFC    60        RTS
;符号判断子程序
EBFD    AD0A04    LDA   $040A
EC00    CD1A04    CMP   $041A
EC03    D00B      BNE   $EC10
EC05    AD0A04    LDA   $040A
EC08    D002      BNE   $EC0C
EC0A    A920      LDA   #$20
EC0C    8D3A04    STA   $043A
EC0F    60        RTS
EC10    A9FF      LDA   #$FF
EC12    60        RTS
;运算结果符号确定子程序
EC13    AD0A04    LDA   $040A
EC16    CD1A04    CMP   $041A
EC19    D018      BNE   $EC33
EC1B    C92D      CMP   #$2D
EC1D    D009      BNE   $EC28
EC1F    8D0A04    STA   $040A
EC22    A920      LDA   #$20
EC24    8D1A04    STA   $041A
EC27    60        RTS
EC28    A920      LDA   #$20
EC2A    8D0A04    STA   $040A
EC2D    A92D      LDA   #$2D
EC2F    8D1A04    STA   $041A
EC32    60        RTS
EC33    AD0A04    LDA   $040A
EC36    8D3A04    STA   $043A
EC39    A9FF      LDA   #$FF
EC3B    60        RTS
EC3C    A220      LDX   #$20
EC3E    AD0A04    LDA   $040A
EC41    CD1A04    CMP   $041A
EC44    F002      BEQ   $EC48
EC46    A22D      LDX   #$2D
EC48    8E3A04    STX   $043A
EC4B    60        RTS
;运算寄存器清零子程序
EC4C    8A        TXA
EC4D    48        PHA
EC4E    A211      LDX   #$11
EC50    A900      LDA   #$00
EC52    958E      STA   $8E,X
EC54    CA        DEX
EC55    10FB      BPL   $EC52
EC57    68        PLA
EC58    AA        TAX
EC59    60        RTS
    卡通定义区复位子程序
EC5A    A000      LDY   #$00
EC5C    842E      STY   $2E
EC5E    842F      STY   $2F
EC60    A9F8      LDA   #$F8
EC62    990002    STA   $0200,Y
EC65    88        DEY
EC66    D0FA      BNE   $EC62
EC68    60        RTS
EC69    2080EC    JSR   $EC80
EC6C    4C2DED    JMP   $ED2D
EC6F    A900      LDA   #$00
EC71    8512      STA   $12
EC73    8513      STA   $13
EC75    204CEE    JSR   $EE4C
EC78    A910      LDA   #$10
EC7A    8D0020    STA   $2000
EC7D    8510      STA   $10
EC7F    60        RTS
EC80    A900      LDA   #$00
EC82    8D0120    STA   $2001
EC85    60        RTS
    卡通定义系列子程序
EC86    A000      LDY   #$00
EC88    B100      LDA   ($00),Y
EC8A    C9FF      CMP   #$FF
EC8C    F013      BEQ   $ECA1
```

```
EC8E    8502      STA    $02
EC90    C8        INY
EC91    B100      LDA    ($00),Y
EC93    8503      STA    $03
EC95    C8        INY
EC96    B100      LDA    ($00),Y
EC98    8504      STA    $04
;定义卡通状态
EC9A    20A2EC    JSR    $ECA2
EC9D    C8        INY
EC9E    4C88EC    JMP    $EC88
ECA1    60        RTS
;定义卡通状态子程序
ECA2    98        TYA
ECA3    48        PHA
ECA4    A902      LDA    #$02
ECA6    8505      STA    $05
ECA8    A603      LDX    $03
ECAA    A000      LDY    #$00
ECAC    A900      LDA    #$00
ECAE    9104      STA    ($04),Y
ECB0    C8        INY
ECB1    C8        INY
ECB2    A502      LDA    $02
ECB4    9104      STA    ($04),Y
ECB6    C8        INY
ECB7    C8        INY
ECB8    CA        DEX
ECB9    D0F1      BNE    $ECAC
ECBB    68        PLA
ECBC    A8        TAY
ECBD    60        RTS
;定义卡通座标、序号的处理
ECBE    8A        TXA
ECBF    48        PHA
ECC0    98        TYA
ECC1    48        PHA
ECC2    A902      LDA    #$02
ECC4    8505      STA    $05
ECC6    A90F      LDA    #$0F
ECC8    2503      AND    $03
ECCA    8507      STA    $07
ECCC    A503      LDA    $03
ECCE    4A        LSR
ECCF    4A        LSR
ECD0    4A        LSR
ECD1    4A        LSR
ECD2    8506      STA    $06
ECD4    AA        TAX
ECD5    A900      LDA    #$00
ECD7    18        CLC
ECD8    6507      ADC    $07
ECDA    CA        DEX
ECDB    D0FB      BNE    $ECD8
ECDD    8508      STA    $08
ECDF    20EAEC    JSR    $ECEA
ECE2    2005ED    JSR    $ED05
ECE5    68        PLA
ECE6    A8        TAY
ECE7    68        PLA
ECE8    AA        TAX
ECE9    60        RTS
;定义卡通序号子程序
ECEA    A502      LDA    $02
ECEC    A608      LDX    $08
ECEE    A001      LDY    #$01
ECF0    9104      STA    ($04),Y
ECF2    18        CLC
ECF3    6901      ADC    #$01
ECF5    C8        INY
ECF6    48        PHA
ECF7    B104      LDA    ($04),Y
ECF9    293F      AND    #$3F
ECFB    9104      STA    ($04),Y
ECFD    68        PLA
ECFE    C8        INY
ECFF    C8        INY
ED00    C8        INY
ED01    CA        DEX
ED02    D0EC      BNE    $ECF0
ED04    60        RTS
;定义卡通座标子程序
ED05    A000      LDY    #$00
ED07    A606      LDX    $06
ED09    A500      LDA    $00
ED0B    8509      STA    $09
ED0D    A501      LDA    $01
ED0F    9104      STA    ($04),Y
```

```
ED11    C8        INY
ED12    C8        INY
ED13    C8        INY
ED14    A509      LDA    $09
ED16    9104      STA    ($04),Y
ED18    18        CLC
ED19    6908      ADC    #$08
ED1B    8509      STA    $09
ED1D    C8        INY
ED1E    CA        DEX
ED1F    D0EC      BNE    $ED0D
ED21    A501      LDA    $01
ED23    18        CLC
ED24    6908      ADC    #$08
ED26    8501      STA    $01
ED28    C607      DEC    $07
ED2A    D0DB      BNE    $ED07
ED2C    60        RTS
```

清零页及配色区子程序

```
ED2D    AD0220    LDA    $2002
ED30    A510      LDA    $10
ED32    29FB      AND    #$FB
ED34    8D0020    STA    $2000
ED37    A920      LDA    #$20
ED39    8D0620    STA    $2006
ED3C    A900      LDA    #$00
ED3E    8D0620    STA    $2006
ED41    A204      LDX    #$04
ED43    A000      LDY    #$00
ED45    A920      LDA    #$20
ED47    8D0720    STA    $2007
ED4A    88        DEY
ED4B    D0FA      BNE    $ED47
ED4D    CA        DEX
ED4E    D0F7      BNE    $ED47
ED50    A923      LDA    #$23
ED52    8D0620    STA    $2006
ED55    A9C0      LDA    #$C0
ED57    8D0620    STA    $2006
ED5A    A040      LDY    #$40
ED5C    A900      LDA    #$00
ED5E    8D0720    STA    $2007
ED61    88        DEY
ED62    D0FA      BNE    $ED5E
ED64    60        RTS
```

对话工作系统送显子程序

```
ED65    8A        TXA
ED66    48        PHA
ED67    98        TYA
ED68    48        PHA
ED69    A000      LDY    #$00
ED6B    B102      LDA    ($02),Y
ED6D    290F      AND    #$0F
ED6F    8505      STA    $05
ED71    B102      LDA    ($02),Y
ED73    4A        LSR
ED74    4A        LSR
ED75    4A        LSR
ED76    4A        LSR
ED77    8504      STA    $04
ED79    AE0003    LDX    $0300
ED7C    A501      LDA    $01
ED7E    9D0103    STA    $0301,X
ED81    20C1ED    JSR    $EDC1
ED84    A500      LDA    $00
ED86    9D0103    STA    $0301,X
ED89    20C1ED    JSR    $EDC1
ED8C    A505      LDA    $05
ED8E    8506      STA    $06
ED90    9D0103    STA    $0301,X
ED93    20C1ED    JSR    $EDC1
ED96    C8        INY
ED97    B102      LDA    ($02),Y
ED99    9D0103    STA    $0301,X
ED9C    C606      DEC    $06
ED9E    D0F3      BNE    $ED93
EDA0    20C1ED    JSR    $EDC1
EDA3    8E0003    STX    $0300
EDA6    18        CLC
EDA7    A920      LDA    #$20
EDA9    6500      ADC    $00
EDAB    8500      STA    $00
EDAD    A900      LDA    #$00
EDAF    6501      ADC    $01
EDB1    8501      STA    $01
EDB3    C604      DEC    $04
EDB5    D0C5      BNE    $ED7C
EDB7    A900      LDA    #$00
```

```
EDB9    9D0103    STA    $0301,X
EDBC    68        PLA
EDBD    A8        TAY
EDBE    68        PLA
EDBF    AA        TAX
EDC0    60        RTS
EDC1    E8        INX
EDC2    8A        TXA
EDC3    C93F      CMP    #$3F
EDC5    900A      BCC    $EDD1
EDC7    AE0003    LDX    $0300
EDCA    A900      LDA    #$00
EDCC    9D0103    STA    $0301,X
EDCF    68        PLA
EDD0    68        PLA
EDD1    60        RTS
```

延时计数子程序

```
EDD2    A209      LDX    #$09
EDD4    C620      DEC    $20
EDD6    1006      BPL    $EDDE
EDD8    A914      LDA    #$14
EDDA    8520      STA    $20
EDDC    A20C      LDX    #$0C
EDDE    B521      LDA    $21,X
EDE0    F002      BEQ    $EDE4
EDE2    D621      DEC    $21,X
EDE4    CA        DEX
EDE5    10F7      BPL    $EDDE
EDE7    60        RTS
```

随机数产生子程序

```
EDE8    A551      LDA    $51
EDEA    2902      AND    #$02
EDEC    8500      STA    $00
EDEE    A552      LDA    $52
EDF0    2902      AND    #$02
EDF2    4500      EOR    $00
EDF4    18        CLC
EDF5    F001      BEQ    $EDF8
EDF7    38        SEC
EDF8    6651      ROR    $51
EDFA    6652      ROR    $52
EDFC    6653      ROR    $53
EDFE    6654      ROR    $54
EE00    6655      ROR    $55
EE02    6656      ROR    $56
EE04    6657      ROR    $57
EE06    6658      ROR    $58
EE08    60        RTS
```

向 PPU 置数子程序

```
EE09    8D0620    STA    $2006
EE0C    C8        INY
EE0D    B100      LDA    ($00),Y
EE0F    8D0620    STA    $2006
EE12    C8        INY
EE13    B100      LDA    ($00),Y
EE15    0A        ASL
EE16    48        PHA
EE17    A510      LDA    $10
EE19    0904      ORA    #$04
EE1B    B002      BCS    $EE1F
EE1D    29FB      AND    #$FB
EE1F    207AEC    JSR    $EC7A
EE22    68        PLA
EE23    0A        ASL
EE24    9003      BCC    $EE29
EE26    0902      ORA    #$02
EE28    C8        INY
EE29    4A        LSR
EE2A    4A        LSR
EE2B    AA        TAX
EE2C    B001      BCS    $EE2F
EE2E    C8        INY
EE2F    B100      LDA    ($00),Y
EE31    8D0720    STA    $2007
EE34    CA        DEX
EE35    D0F5      BNE    $EE2C
EE37    38        SEC
EE38    98        TYA
EE39    6500      ADC    $00
EE3B    8500      STA    $00
EE3D    A900      LDA    #$00
EE3F    6501      ADC    $01
EE41    8501      STA    $01
EE43    AE0220    LDX    $2002
EE46    A000      LDY    #$00
EE48    B100      LDA    ($00),Y
EE4A    D0BD      BNE    $EE09
EE4C    A512      LDA    $12
```

```
EE4E   8D0520   STA   $2005
EE51   A513     LDA   $13
EE53   8D0520   STA   $2005
EE56   60       RTS
```

OPERATOR 栏操作使用的数据区

```
EE57                                11 20 00 01 00 00 01 BF 97
EE60   D7 11 67 18 18 18 00 18 3D 00 00 00 00 3C 3C 3C
EE70   97 3C AC BC BC 97 AC DA DA DA C8 DA 22 22 22 38
EE80   22 10 08 10 00 08 20 20 20 22 20 E4 E4 E4 67 E4
EE90   0C 0C 0C 09 0C 03 03 03 04 03 34 14 34 00 14 28
EEA0   08 28 00 08 40 20 40 00 20 FF FF FF FF 01 01 01
EEB0   01 01 01 01 01 FF FF FF FF FF FF 01 01 01 01 01
EEC0   01 FF FF FF FF FF FF 01 01 00 00 01 FF 48 A8 08
EED0   60 C0 20 C1 C2 C3 C4 C5 C6 C7 C8 2B 2D B5 B6 28
EEE0   29 2E 3D 63 64 65 66 67 61 62 C8 02 06 0E 0F 10
EEF0   11 17 18 43 44 45 46 47 41 42 24
```

OPERATOR 栏操作预处理

```
EEFB   A920     LDA   #$20
EEFD   A000     LDY   #$00
EEFF   993D06   STA   $063D,Y
EF02   88       DEY
EF03   D0FA     BNE   $EEFF
EF05   A097     LDY   #$97
EF07   993C07   STA   $073C,Y
EF0A   88       DEY
EF0B   D0FA     BNE   $EF07
EF0D   A636     LDX   $36
EF0F   BD7CEE   LDA   $EE7C,X
EF12   8530     STA   $30
EF14   BD6DEE   LDA   $EE6D,X
EF17   8531     STA   $31
EF19   BD86EE   LDA   $EE86,X
EF1C   8532     STA   $32
EF1E   BD8BEE   LDA   $EE8B,X
EF21   8534     STA   $34
EF23   A900     LDA   #$00
EF25   852E     STA   $2E
EF27   852F     STA   $2F
EF29   8542     STA   $42
EF2B   60       RTS
```

OPERATOR 栏有字符输入的处理

```
EF2C   8636     STX   $36
EF2E   E003     CPX   #$03
EF30   F029     BEQ   $EF5B
EF32   A523     LDA   $23
EF34   F025     BEQ   $EF5B
EF36   C901     CMP   #$01
EF38   F01A     BEQ   $EF54
EF3A   A537     LDA   $37
EF3C   C539     CMP   $39
EF3E   F010     BEQ   $EF50
EF40   A539     LDA   $39
EF42   F007     BEQ   $EF4B
EF44   A537     LDA   $37
EF46   8539     STA   $39
EF48   4C50EF   JMP   $EF50
EF4B   A902     LDA   #$02
EF4D   8523     STA   $23
EF4F   60       RTS
;羽毛笔画圈处理
EF50   20E1F4   JSR   $F4E1
EF53   60       RTS
EF54   A900     LDA   #$00
EF56   853C     STA   $3C
EF58   4C76F0   JMP   $F076
EF5B   A900     LDA   #$00
EF5D   853A     STA   $3A
EF5F   85B0     STA   $B0
EF61   8540     STA   $40
EF63   A537     LDA   $37
EF65   8538     STA   $38
EF67   C539     CMP   $39
EF69   D01A     BNE   $EF85
EF6B   A538     LDA   $38
EF6D   F00B     BEQ   $EF7A
```

```
EF6F    A521      LDA   $21
EF71    D006      BNE   $EF79
EF73    8539      STA   $39
EF75    A908      LDA   #$08
EF77    8521      STA   $21
EF79    60        RTS
EF7A    A920      LDA   #$20
EF7C    8521      STA   $21
EF7E    A58B      LDA   $8B
EF80    D0F7      BNE   $EF79
EF82    4C90F0    JMP   $F090
     输入新键值的判断处理
EF85    A538      LDA   $38
EF87    8539      STA   $39
EF89    F0EF      BEQ   $EF7A
EF8B    8540      STA   $40
EF8D    A58B      LDA   $8B
EF8F    D0E8      BNE   $EF79
EF91    A538      LDA   $38
EF93    C91C      CMP   #$1C
EF95    D009      BNE   $EFA0
;→键的处理
EF97    2010F2    JSR   $F210
EF9A    2015F1    JSR   $F115
EF9D    4C90F0    JMP   $F090
EFA0    C91D      CMP   #$1D
EFA2    D009      BNE   $EFAD
;←键的处理
EFA4    203FF2    JSR   $F23F
EFA7    2084F1    JSR   $F184
EFAA    4C90F0    JMP   $F090
EFAD    C91E      CMP   #$1E
EFAF    D012      BNE   $EFC3
;↑键的处理
EFB1    A531      LDA   $31
EFB3    DD6DEE    CMP   $EE6D,X
EFB6    F008      BEQ   $EFC0
EFB8    2066F2    JSR   $F266
EFBB    20F7F1    JSR   $F1F7
EFBE    C642      DEC   $42
EFC0    4C90F0    JMP   $F090
EFC3    C91F      CMP   #$1F
EFC5    D012      BNE   $EFD9
;↓键的处理
EFC7    A531      LDA   $31
EFC9    DD72EE    CMP   $EE72,X
EFCC    F008      BEQ   $EFD6
EFCE    2059F2    JSR   $F259
EFD1    20DEF1    JSR   $F1DE
EFD4    E642      INC   $42
EFD6    4C90F0    JMP   $F090
EFD9    C90D      CMP   #$0D
EFDB    D01B      BNE   $EFF8
;回车键的处理
EFDD    2091F2    JSR   $F291
EFE0    A920      LDA   #$20
EFE2    8523      STA   $23
EFE4    A904      LDA   #$04
EFE6    85FE      STA   $FE
EFE8    A9D6      LDA   #$D6
EFEA    8DC302    STA   $02C3
EFED    A9A4      LDA   #$A4
EFEF    8DC002    STA   $02C0
EFF2    853C      STA   $3C
EFF4    20E1F4    JSR   $F4E1
EFF7    60        RTS
;功能键的处理
EFF8    C9C1      CMP   #$C1
EFFA    9005      BCC   $F001
EFFC    C9C9      CMP   #$C9
EFFE    B001      BCS   $F001
F000    60        RTS
F001    A538      LDA   $38
F003    C908      CMP   #$08
F005    D013      BNE   $F01A
;Del 键的处理
F007    AD0003    LDA   $0300
F00A    D0F4      BNE   $F000
F00C    20D4F0    JSR   $F0D4
F00F    D0EF      BNE   $F000
F011    2073F4    JSR   $F473
F014    20A9F2    JSR   $F2A9
F017    4C90F0    JMP   $F090
F01A    C912      CMP   #$12
F01C    D013      BNE   $F031
;Ins 键的处理
F01E    AD0003    LDA   $0300
F021    D0DD      BNE   $F000
```

```
F023   20D4F0   JSR   $F0D4
F026   D0D8     BNE   $F000
F028   2073F4   JSR   $F473
F02B   204BF3   JSR   $F34B
F02E   4C90F0   JMP   $F090
F031   C920     CMP   #$20
F033   B009     BCS   $F03E
;键值<H20的处理
F035   E004     CPX   #$04
F037   D060     BNE   $F099
F039   20C8F4   JSR   $F4C8
F03C   F05B     BEQ   $F099
;键值≥H20的处理
F03E   A8       TAY
F03F   20D4F0   JSR   $F0D4
F042   D046     BNE   $F08A
F044   A62F     LDX   $2F
F046   A52E     LDA   $2E
F048   F007     BEQ   $F051
F04A   98       TYA
F04B   9D3D07   STA   $073D,X
F04E   4C55F0   JMP   $F055
F051   98       TYA
F052   9D3D06   STA   $063D,X
F055   C920     CMP   #$20
F057   F01D     BEQ   $F076
F059   A536     LDA   $36
F05B   C903     CMP   #$03
F05D   F013     BEQ   $F072
F05F   C904     CMP   #$04
F061   D006     BNE   $F069
;MUSIC板的处理
F063   20B5F0   JSR   $F0B5
F066   4C6DF0   JMP   $F06D
F069   A940     LDA   #$40
F06B   85FE     STA   $FE
F06D   A910     LDA   #$10
F06F   8523     STA   $23
F071   60       RTS
;对话板的处理
F072   A904     LDA   #$04
F074   85FF     STA   $FF
F076   A538     LDA   $38
F078   C90D     CMP   #$0D
F07A   F01D     BEQ   $F099
F07C   2088F2   JSR   $F288
F07F   A538     LDA   $38
F081   A636     LDX   $36
F083   DD68EE   CMP   $EE68,X
F086   F012     BEQ   $F09A
F088   E63A     INC   $3A
F08A   2010F2   JSR   $F210
F08D   2015F1   JSR   $F115
F090   A536     LDA   $36
F092   C903     CMP   #$03
F094   F003     BEQ   $F099
F096   200FF5   JSR   $F50F
F099   60       RTS
        CAL.板按==键的处理
F09A   E6B0     INC   $B0
F09C   A62F     LDX   $2F
F09E   BD3D06   LDA   $063D,X
F0A1   C90D     CMP   #$0D
F0A3   F00A     BEQ   $F0AF
F0A5   C920     CMP   #$20
F0A7   F006     BEQ   $F0AF
F0A9   E8       INX
F0AA   E0D9     CPX   #$D9
F0AC   90F0     BCC   $F09E
F0AE   60       RTS
F0AF   A920     LDA   #$20
F0B1   9D3D06   STA   $063D,X
F0B4   60       RTS
        MUSIC板的演奏处理
F0B5   A542     LDA   $42
F0B7   2903     AND   #$03
F0B9   A8       TAY
F0BA   A538     LDA   $38
F0BC   C000     CPY   #$00
F0BE   D004     BNE   $F0C4
F0C0   20CFE5   JSR   $E5CF
F0C3   60       RTS
F0C4   C001     CPY   #$01
F0C6   D004     BNE   $F0CC
F0C8   20F4E5   JSR   $E5F4
F0CB   60       RTS
F0CC   C002     CPY   #$02
F0CE   D003     BNE   $F0D3
```

```
F0D0   2019E6   JSR   $E619
F0D3   60       RTS
F0D4   A536     LDA   $36
F0D6   C904     CMP   #$04
F0D8   D01D     BNE   $F0F7
F0DA   A200     LDX   #$00
F0DC   BDC9EE   LDA   $EEC9,X
F0DF   C9FF     CMP   #$FF
F0E1   F014     BEQ   $F0F7
F0E3   C52E     CMP   $2E
F0E5   D00C     BNE   $F0F3
F0E7   A52F     LDA   $2F
F0E9   DDCDEE   CMP   $EECD,X
F0EC   9005     BCC   $F0F3
F0EE   DDD0EE   CMP   $EED0,X
F0F1   9007     BCC   $F0FA
F0F3   E8       INX
F0F4   4CDCF0   JMP   $F0DC
F0F7   A900     LDA   #$00
F0F9   60       RTS
F0FA   A901     LDA   #$01
F0FC   60       RTS
F0FD   A62F     LDX   $2F
F0FF   E000     CPX   #$00
F101   F00C     BEQ   $F10F
F103   CA       DEX
F104   BD3D06   LDA   $063D,X
F107   C93D     CMP   #$3D
F109   F007     BEQ   $F112
F10B   C90D     CMP   #$0D
F10D   D0F0     BNE   $F0FF
F10F   A900     LDA   #$00
F111   60       RTS
F112   A901     LDA   #$01
F114   60       RTS
```

调整 OPERATOR 栏的字符显示地址和光标、羽毛笔的座标

```
F115   8A       TXA
F116   48       PHA
F117   A636     LDX   $36
F119   BD59EE   LDA   $EE59,X
F11C   C52E     CMP   $2E
F11E   D01D     BNE   $F13D
F120   A52F     LDA   $2F
F122   DD5EEE   CMP   $EE5E,X
F125   B054     BCS   $F17B
F127   E004     CPX   #$04
F129   D012     BNE   $F13D
F12B   A52E     LDA   $2E
F12D   F00E     BEQ   $F13D
F12F   A52F     LDA   $2F
F131   C91F     CMP   #$1F
F133   F046     BEQ   $F17B
F135   C937     CMP   #$37
F137   F042     BEQ   $F17B
F139   C94F     CMP   #$4F
F13B   F03E     BEQ   $F17B
F13D   2073F2   JSR   $F273
F140   A530     LDA   $30
F142   DD77EE   CMP   $EE77,X
F145   D021     BNE   $F168
F147   BD7CEE   LDA   $EE7C,X
F14A   8530     STA   $30
F14C   E004     CPX   #$04
F14E   D012     BNE   $F162
```

;MUSIC 板的输入换行处理

```
F150   20DEF1   JSR   $F1DE
F153   20DEF1   JSR   $F1DE
F156   20DEF1   JSR   $F1DE
F159   2059F2   JSR   $F259
F15C   2059F2   JSR   $F259
F15F   2059F2   JSR   $F259
F162   20EEF1   JSR   $F1EE
F165   4C6FF1   JMP   $F16F
F168   A530     LDA   $30
F16A   18       CLC
F16B   6908     ADC   #$08
F16D   8530     STA   $30
F16F   E004     CPX   #$04
F171   D00A     BNE   $F17D
F173   E634     INC   $34
F175   2027F4   JSR   $F427
F178   4C7DF1   JMP   $F17D
F17B   A902     LDA   #$02
F17D   8500     STA   $00
F17F   68       PLA
F180   AA       TAX
F181   A500     LDA   $00
```

```
F183    60       RTS
        按←键的处理
F184    A636     LDX    $36
F186    A52E     LDA    $2E
F188    052F     ORA    $2F
F18A    F051     BEQ    $F1DD
F18C    E004     CPX    #$04
F18E    D012     BNE    $F1A2
F190    A52E     LDA    $2E
F192    D00E     BNE    $F1A2
F194    A52F     LDA    $2F
F196    C918     CMP    #$18
F198    F043     BEQ    $F1DD
F19A    C930     CMP    #$30
F19C    F03F     BEQ    $F1DD
F19E    C948     CMP    #$48
F1A0    F03B     BEQ    $F1DD
F1A2    207AF2   JSR    $F27A
F1A5    A530     LDA    $30
F1A7    DD7CEE   CMP    $EE7C,X
F1AA    D021     BNE    $F1CD
;羽毛笔位于行首时上移一行的处理
F1AC    BD77EE   LDA    $EE77,X
F1AF    8530     STA    $30
F1B1    E004     CPX    #$04
F1B3    D012     BNE    $F1C7
F1B5    20F7F1   JSR    $F1F7
F1B8    20F7F1   JSR    $F1F7
F1BB    20F7F1   JSR    $F1F7
F1BE    2066F2   JSR    $F266
F1C1    2066F2   JSR    $F266
F1C4    2066F2   JSR    $F266
F1C7    2007F2   JSR    $F207
F1CA    4CD4F1   JMP    $F1D4
;令羽毛笔前移一格
F1CD    A530     LDA    $30
F1CF    38       SEC
F1D0    E908     SBC    #$08
F1D2    8530     STA    $30
F1D4    E004     CPX    #$04
F1D6    D005     BNE    $F1DD
F1D8    C634     DEC    $34
F1DA    2053F4   JSR    $F453
F1DD    60       RTS
        羽毛笔的换行处理
F1DE    A636     LDX    $36
F1E0    A52F     LDA    $2F
F1E2    18       CLC
;令羽毛笔下移
F1E3    7D63EE   ADC    $EE63,X
F1E6    852F     STA    $2F
F1E8    A52E     LDA    $2E
F1EA    6900     ADC    #$00
F1EC    852E     STA    $2E
F1EE    A531     LDA    $31
F1F0    18       CLC
F1F1    7D81EE   ADC    $EE81,X
F1F4    8531     STA    $31
F1F6    60       RTS
;令羽毛笔上移
F1F7    A636     LDX    $36
F1F9    A52F     LDA    $2F
F1FB    38       SEC
F1FC    FD63EE   SBC    $EE63,X
F1FF    852F     STA    $2F
F201    A52E     LDA    $2E
F203    E900     SBC    #$00
F205    852E     STA    $2E
F207    A531     LDA    $31
F209    38       SEC
F20A    FD81EE   SBC    $EE81,X
F20D    8531     STA    $31
F20F    60       RTS
        调整OPERATOR栏或MUSIC
        栏的显示地址
F210    8A       TXA
F211    48       PHA
F212    98       TYA
F213    48       PHA
F214    A636     LDX    $36
F216    BD59EE   LDA    $EE59,X
F219    A479     LDY    $79
F21B    D004     BNE    $F221
F21D    E004     CPX    #$04
F21F    F015     BEQ    $F236
F221    C52E     CMP    $2E
F223    D007     BNE    $F22C
F225    A52F     LDA    $2F
```

```
F227    DD5EEE    CMP    $EE5E,X
F22A    B008      BCS    $F234
F22C    E634      INC    $34
F22E    2027F4    JSR    $F427
F231    4C36F2    JMP    $F236
F234    A902      LDA    #$02
F236    8500      STA    $00
F238    68        PLA
F239    A8        TAY
F23A    68        PLA
F23B    AA        TAX
F23C    A500      LDA    $00
F23E    60        RTS
```

按 Del 键时检查羽毛笔的座标

```
F23F    8A        TXA
F240    48        PHA
F241    98        TYA
F242    48        PHA
F243    A636      LDX    $36
F245    A52E      LDA    $2E
F247    052F      ORA    $2F
F249    F009      BEQ    $F254
F24B    E004      CPX    #$04
F24D    F0E5      BEQ    $F234
F24F    C634      DEC    $34
F251    2053F4    JSR    $F453
F254    68        PLA
F255    A8        TAY
F256    68        PLA
F257    AA        TAX
F258    60        RTS
```

按↓键改变送显字符的地址

```
F259    A534      LDA    $34
F25B    18        CLC
F25C    7DA4EE    ADC    $EEA4,X
F25F    8534      STA    $34
F261    9002      BCC    $F265
F263    E632      INC    $32
F265    60        RTS
```

按↑键改变送显字符的地址

```
F266    A534      LDA    $34
F268    38        SEC
F269    FDA4EE    SBC    $EEA4,X
F26C    8534      STA    $34
F26E    B002      BCS    $F272
F270    C632      DEC    $32
F272    60        RTS
```

按→键和输入字符时令 $2F+1

```
F273    E62F      INC    $2F
F275    D002      BNE    $F279
F277    E62E      INC    $2E
F279    60        RTS
```

按←键和删去字符时令 $2F-1

```
F27A    A52F      LDA    $2F
F27C    38        SEC
F27D    E901      SBC    #$01
F27F    852F      STA    $2F
F281    A52E      LDA    $2E
F283    E900      SBC    #$00
F285    852E      STA    $2E
F287    60        RTS
F288    2002F4    JSR    $F402
F28B    A538      LDA    $38
F28D    8D0403    STA    $0304
F290    60        RTS
```

MUSIC 板按回车键的处理

```
F291    A636      LDX    $36
F293    2015F1    JSR    $F115
F296    E004      CPX    #$04
F298    F003      BEQ    $F29D
F29A    2010F2    JSR    $F210
F29D    C900      CMP    #$00
F29F    F0F2      BEQ    $F293
F2A1    C902      CMP    #$02
F2A3    D003      BNE    $F2A8
F2A5    200DEF    JSR    $EF0D
F2A8    60        RTS
```

按 Del 键的处理

```
F2A9    A636      LDX    $36
F2AB    A530      LDA    $30
F2AD    DD7CEE    CMP    $EE7C,X
F2B0    F006      BEQ    $F2B8
```

;令羽毛笔后的各字符前移一格

```
F2B2    203FF2    JSR    $F23F
F2B5    2084F1    JSR    $F184
F2B8    A52F      LDA    $2F
F2BA    8509      STA    $09
F2BC    A52E      LDA    $2E
```

```
F2BE    850A      STA   $0A
F2C0    2088F2    JSR   $F288
F2C3    A532      LDA   $32
F2C5    8507      STA   $07
F2C7    A534      LDA   $34
F2C9    8508      STA   $08
F2CB    A201      LDX   #$01
F2CD    A42F      LDY   $2F
F2CF    A52E      LDA   $2E
F2D1    D00D      BNE   $F2E0
F2D3    B93E06    LDA   $063E,Y
F2D6    C0FF      CPY   #$FF
F2D8    D003      BNE   $F2DD
F2DA    AD3D07    LDA   $073D
F2DD    4CE3F2    JMP   $F2E3
F2E0    B93E07    LDA   $073E,Y
F2E3    9D0303    STA   $0303,X
F2E6    850B      STA   $0B
F2E8    E8        INX
F2E9    E634      INC   $34
F2EB    8A        TXA
F2EC    48        PHA
F2ED    98        TYA
F2EE    48        PHA
;检查是否达到右边界
F2EF    2027F4    JSR   $F427
F2F2    68        PLA
F2F3    A8        TAY
F2F4    68        PLA
F2F5    AA        TAX
F2F6    A5BB      LDA   $BB
F2F8    D019      BNE   $F313
F2FA    A52E      LDA   $2E
F2FC    D008      BNE   $F306
F2FE    A50B      LDA   $0B
F300    993D06    STA   $063D,Y
F303    4C0BF3    JMP   $F30B
F306    A50B      LDA   $0B
F308    993D07    STA   $073D,Y
F30B    2073F2    JSR   $F273
F30E    A42F      LDY   $2F
F310    4CCFF2    JMP   $F2CF
F313    A920      LDA   #$20
F315    9D0203    STA   $0302,X
F318    A52E      LDA   $2E
F31A    D008      BNE   $F324
F31C    A920      LDA   #$20
F31E    993D06    STA   $063D,Y
F321    4C29F3    JMP   $F329
F324    A920      LDA   #$20
F326    993D07    STA   $073D,Y
F329    9D0303    STA   $0303,X
F32C    8E0303    STX   $0303
F32F    E8        INX
F330    A900      LDA   #$00
F332    9D0303    STA   $0303,X
F335    E8        INX
F336    E8        INX
F337    8E0003    STX   $0300
F33A    A507      LDA   $07
F33C    8532      STA   $32
F33E    A508      LDA   $08
F340    8534      STA   $34
F342    A509      LDA   $09
F344    852F      STA   $2F
F346    A50A      LDA   $0A
F348    852E      STA   $2E
F34A    60        RTS
```

按 Ins 键的处理

```
;令羽毛笔后面移出一个空格,当最后的字符移到画面右边界时,若继续按 Ins 键,则令最后一个字符消失
F34B    A532      LDA   $32
F34D    8507      STA   $07
F34F    A534      LDA   $34
F351    8508      STA   $08
F353    A42F      LDY   $2F
F355    8409      STY   $09
F357    2088F2    JSR   $F288
F35A    A201      LDX   #$01
F35C    A52E      LDA   $2E
F35E    850B      STA   $0B
F360    D00C      BNE   $F36E
F362    B93D06    LDA   $063D,Y
F365    48        PHA
F366    A920      LDA   #$20
F368    993D06    STA   $063D,Y
F36B    4C77F3    JMP   $F377
```

```
F36E   B93D07   LDA   $073D,Y
F371   48       PHA
F372   A920     LDA   #$20
F374   993D07   STA   $073D,Y
F377   9D0303   STA   $0303,X
F37A   68       PLA
F37B   850A     STA   $0A
F37D   E634     INC   $34
F37F   8A       TXA
F380   48       PHA
F381   98       TYA
F382   48       PHA
F383   2027F4   JSR   $F427
F386   68       PLA
F387   A8       TAY
F388   68       PLA
F389   AA       TAX
F38A   A5BB     LDA   $BB
F38C   D046     BNE   $F3D4
F38E   A508     LDA   $08
F390   8534     STA   $34
F392   A52E     LDA   $2E
F394   D006     BNE   $F39C
F396   B93E06   LDA   $063E,Y
F399   4C9FF3   JMP   $F39F
F39C   B93E07   LDA   $073E,Y
F39F   48       PHA
F3A0   E8       INX
F3A1   A50A     LDA   $0A
F3A3   9D0303   STA   $0303,X
F3A6   E634     INC   $34
F3A8   8A       TXA
F3A9   48       PHA
F3AA   98       TYA
F3AB   48       PHA
F3AC   2027F4   JSR   $F427
F3AF   68       PLA
F3B0   A8       TAY
F3B1   68       PLA
F3B2   AA       TAX
F3B3   A5BB     LDA   $BB
F3B5   D01C     BNE   $F3D3
F3B7   A52E     LDA   $2E
F3B9   D008     BNE   $F3C3
F3BB   A50A     LDA   $0A
F3BD   993E06   STA   $063E,Y
F3C0   4CC8F3   JMP   $F3C8
F3C3   A50A     LDA   $0A
F3C5   993E07   STA   $073E,Y
F3C8   68       PLA
F3C9   850A     STA   $0A
F3CB   2073F2   JSR   $F273
F3CE   A42F     LDY   $2F
F3D0   4C92F3   JMP   $F392
F3D3   68       PLA
F3D4   A920     LDA   #$20
F3D6   9D0303   STA   $0303,X
F3D9   8E0303   STX   $0303
F3DC   E8       INX
F3DD   A900     LDA   #$00
F3DF   9D0303   STA   $0303,X
F3E2   E8       INX
F3E3   E8       INX
F3E4   8E0003   STX   $0300
F3E7   A507     LDA   $07
F3E9   8532     STA   $32
F3EB   A508     LDA   $08
F3ED   8534     STA   $34
F3EF   A509     LDA   $09
F3F1   852F     STA   $2F
F3F3   A50B     LDA   $0B
F3F5   852E     STA   $2E
F3F7   60       RTS
F3F8   8A       TXA
F3F9   48       PHA
F3FA   A201     LDX   #$01
F3FC   2004F4   JSR   $F404
F3FF   68       PLA
F400   AA       TAX
F401   60       RTS
```

显示操作者的输入内容

```
F402   A200     LDX   #$00
F404   B534     LDA   $34,X
F406   8500     STA   $00
F408   B532     LDA   $32,X
F40A   8501     STA   $01
F40C   A957     LDA   #$57
F40E   8502     STA   $02
```

```
F410    A9EE      LDA   #$EE
F412    8503      STA   $03
F414    4C65ED    JMP   $ED65
F417    8A        TXA
F418    48        PHA
F419    98        TYA
F41A    48        PHA
F41B    A201      LDX   #$01
F41D    E635      INC   $35
F41F    2029F4    JSR   $F429
F422    68        PLA
F423    A8        TAY
F424    68        PLA
F425    AA        TAX
F426    60        RTS
```

检查OPERATOR栏送显字符的数量，调整显示地址

```
F427    A200      LDX   #$00
F429    A900      LDA   #$00
F42B    85BB      STA   $BB
F42D    A436      LDY   $36
F42F    B534      LDA   $34,X
F431    2910      AND   #$10
F433    F01B      BEQ   $F450
F435    B534      LDA   $34,X
F437    290F      AND   #$0F
F439    D990EE    CMP   $EE90,Y
F43C    D012      BNE   $F450
F43E    E6BB      INC   $BB
F440    B534      LDA   $34,X
F442    29F0      AND   #$F0
F444    18        CLC
F445    799AEE    ADC   $EE9A,Y
F448    9534      STA   $34,X
F44A    B532      LDA   $32,X
F44C    6900      ADC   #$00
F44E    9532      STA   $32,X
F450    A5BB      LDA   $BB
F452    60        RTS
F453    A436      LDY   $36
F455    A534      LDA   $34
F457    2910      AND   #$10
F459    D017      BNE   $F472
F45B    A534      LDA   $34
F45D    290F      AND   #$0F
F45F    D995EE    CMP   $EE95,Y
F462    D00E      BNE   $F472
F464    A534      LDA   $34
F466    38        SEC
F467    F99FEE    SBC   $EE9F,Y
F46A    8534      STA   $34
F46C    A532      LDA   $32
F46E    E900      SBC   #$00
F470    8532      STA   $32
F472    60        RTS
```

;按Del键时用于检查本行输入的最后字符位置

```
F473    A532      LDA   $32
F475    8533      STA   $33
F477    A534      LDA   $34
F479    8535      STA   $35
F47B    A62F      LDX   $2F
F47D    BD3D06    LDA   $063D,X
F480    C93D      CMP   #$3D
F482    F011      BEQ   $F495
F484    C90D      CMP   #$0D
F486    F023      BEQ   $F4AB
F488    C920      CMP   #$20
F48A    F01F      BEQ   $F4AB
F48C    2017F4    JSR   $F417
F48F    E8        INX
F490    E0D9      CPX   #$D9
F492    90E9      BCC   $F47D
F494    60        RTS
```

;CAL. 板时按"="键的处理

```
F495    A537      LDA   $37
F497    C908      CMP   #$08
F499    F010      BEQ   $F4AB
F49B    C912      CMP   #$12
F49D    F00C      BEQ   $F4AB
F49F    20F8F3    JSR   $F3F8
F4A2    A920      LDA   #$20
F4A4    8D0803    STA   $0308
F4A7    9D3D06    STA   $063D,X
F4AA    60        RTS
F4AB    A920      LDA   #$20
F4AD    9D3D06    STA   $063D,X
F4B0    60        RTS
```

定义功能键的处理

```
F4B1    A200      LDX    #$00
F4B3    A000      LDY    #$00
;根据X的取值,定义八个功能键
F4B5    BDD3EE    LDA    $EED3,X
F4B8    998004    STA    $0480,Y
F4BB    E8        INX
F4BC    98        TYA
F4BD    18        CLC
F4BE    6910      ADC    #$10
F4C0    A8        TAY
F4C1    C080      CPY    #$80
F4C3    90F0      BCC    $F4B5
F4C5    4C13CF    JMP    $CF13
F4C8    A470      LDY    $70
F4CA    F00C      BEQ    $F4D8
F4CC    A000      LDY    #$00
F4CE    D9EBEE    CMP    $EEEB,Y
F4D1    F008      BEQ    $F4DB
F4D3    C8        INY
F4D4    C008      CPY    #$08
F4D6    30F6      BMI    $F4CE
F4D8    A900      LDA    #$00
F4DA    60        RTS
F4DB    B9F3EE    LDA    $EEF3,Y
F4DE    8538      STA    $38
F4E0    60        RTS
    羽毛笔划圈处理
F4E1    A523      LDA    $23
F4E3    290F      AND    #$0F
F4E5    AA        TAX
F4E6    ADC302    LDA    $02C3
F4E9    18        CLC
F4EA    7DA9EE    ADC    $EEA9,X
F4ED    8500      STA    $00
F4EF    ADC002    LDA    $02C0
F4F2    18        CLC
F4F3    7DB9EE    ADC    $EEB9,X
F4F6    8501      STA    $01
F4F8    A53C      LDA    $3C
F4FA    F010      BEQ    $F50C
F4FC    A920      LDA    #$20
F4FE    8521      STA    $21
F500    ADC302    LDA    $02C3
F503    8500      STA    $00
F505    A9E8      LDA    #$E8
F507    8502      STA    $02
F509    4C1FF5    JMP    $F51F
F50C    4C17F5    JMP    $F517
    羽毛笔卡通的定义处理
F50F    A530      LDA    $30
F511    8500      STA    $00
F513    A531      LDA    $31
F515    8501      STA    $01
F517    A9E0      LDA    #$E0
F519    8502      STA    $02
F51B    A900      LDA    #$00
F51D    853C      STA    $3C
F51F    A500      LDA    $00
F521    48        PHA
F522    A501      LDA    $01
F524    48        PHA
F525    A502      LDA    $02
F527    48        PHA
F528    2083D1    JSR    $D183
F52B    A572      LDA    $72
F52D    F003      BEQ    $F532
F52F    208CD1    JSR    $D18C
F532    68        PLA
F533    8502      STA    $02
F535    68        PLA
F536    8501      STA    $01
F538    68        PLA
F539    8500      STA    $00
F53B    A924      LDA    #$24
F53D    8503      STA    $03
F53F    A9C0      LDA    #$C0
F541    8504      STA    $04
F543    4CBEEC    JMP    $ECBE
    控制对话画面OPERATOR栏中光标的闪烁
F546    A53B      LDA    $3B
F548    2910      AND    #$10
F54A    F005      BEQ    $F551
F54C    A9F8      LDA    #$F8
F54E    4C53F5    JMP    $F553
F551    A531      LDA    $31
F553    8DC002    STA    $02C0
F556    A530      LDA    $30
F558    8DC302    STA    $02C3
```

```
F55B   A9DD     LDA   #$DD
F55D   8DC102   STA   $02C1
F560   A572     LDA   $72
F562   F005     BEQ   $F569
F564   A9DC     LDA   #$DC
F566   8DC102   STA   $02C1
F569   A900     LDA   #$00
F56B   8DC202   STA   $02C2
F56E   60       RTS
```

以下程序在系统移植时没有使用，在此列出供参考

```
F56F                                                FF
F570   1D 20 23 00 00 00 04 04 08 00 00 00 00 00 00 04
F580   08 0C 10 14 18 1C 20 24 28 30 30 30 31 31 38 36
F590   38 31 39 31 32 30 30 34 34 30 30 31 34 30 39 39
F5A0   39 32 39 32 34 31 39 31 31 31 39 32 35 30 30 31
F5B0   32 30 30 33 31 30 30 33 30 30 30 32 38 30 30 32
F5C0   39 07 06 08 01 01 01 09 01 05 02 09 01 4D 54 53
F5D0   07 09 07 08 07 08 07 07 08 07 08 07 E2 F5 EE F5
F5E0   FC F5 02 45 67 77 64 30 9B CE FF FE DC A0 02 34
F5F0   56 77 76 54 32 0A BC DE FF FE DC BA 01 34 56 67
F600   77 76 54 32 09 AB CD EF FF FE ED CB 90 00 00 00
F610   00 03 01 00 05 09 00 09 00 01 02 00 01 05 01 01
F620   08 01 02 01 02 02 04 03 02 07 03 03 00 04 03 03
F630   04 23 28 33
```

```
F634   A209     LDX   #$09
F636   B5D0     LDA   $D0,X
F638   95A0     STA   $A0,X
F63A   BD3005   LDA   $0530,X
F63D   9D2005   STA   $0520,X
F640   CA       DEX
F641   10F3     BPL   $F636
F643   4C13CF   JMP   $CF13
F646   A56B     LDA   $6B
F648   8587     STA   $87
F64A   A573     LDA   $73
F64C   F004     BEQ   $F652
F64E   A90D     LDA   #$0D
F650   85AF     STA   $AF
F652   205DD0   JSR   $D05D
F655   F004     BEQ   $F65B
F657   4CF7D0   JMP   $D0F7
F65A   60       RTS
F65B   20C4CF   JSR   $CFC4
F65E   A564     LDA   $64
F660   F0F8     BEQ   $F65A
F662   A500     LDA   $00
F664   C904     CMP   #$04
F666   F009     BEQ   $F671
F668   C905     CMP   #$05
F66A   F0EE     BEQ   $F65A
F66C   A91D     LDA   #$1D
F66E   4C9CD2   JMP   $D29C
F671   A582     LDA   $82
F673   8587     STA   $87
F675   A923     LDA   #$23
F677   4C9CD2   JMP   $D29C
F67A   A573     LDA   $73
F67C   F004     BEQ   $F682
F67E   A920     LDA   #$20
F680   D002     BNE   $F684
F682   A910     LDA   #$10
F684   85FE     STA   $FE
F686   4CD5CE   JMP   $CED5
F689   A573     LDA   $73
F68B   F035     BEQ   $F6C2
F68D   A209     LDX   #$09
F68F   B5D0     LDA   $D0,X
F691   290F     AND   #$0F
F693   9D4004   STA   $0440,X
F696   CA       DEX
F697   10F6     BPL   $F68F
F699   20FBF7   JSR   $F7FB
```

```
F69C   A203     LDX   #$03
F69E   B5A0     LDA   $A0,X
F6A0   9D1504   STA   $0415,X
F6A3   CA       DEX
F6A4   10F8     BPL   $F69E
F6A6   204BE7   JSR   $E74B
F6A9   AD3A04   LDA   $043A
F6AC   C92D     CMP   #$2D
F6AE   F04D     BEQ   $F6FD
F6B0   AD3504   LDA   $0435
F6B3   290F     AND   #$0F
F6B5   D007     BNE   $F6BE
F6B7   AD3604   LDA   $0436
F6BA   290F     AND   #$0F
F6BC   F007     BEQ   $F6C5
F6BE   A9FF     LDA   #$FF
F6C0   D03D     BNE   $F6FF
F6C2   4C6FF7   JMP   $F76F
F6C5   200AF8   JSR   $F80A
F6C8   A208     LDX   #$08
F6CA   BD3004   LDA   $0430,X
F6CD   9D4904   STA   $0449,X
F6D0   CA       DEX
F6D1   10F7     BPL   $F6CA
F6D3   A208     LDX   #$08
F6D5   B5A0     LDA   $A0,X
F6D7   290F     AND   #$0F
F6D9   9D4004   STA   $0440,X
F6DC   CA       DEX
F6DD   10F6     BPL   $F6D5
F6DF   200AF8   JSR   $F80A
F6E2   20B4F7   JSR   $F7B4
F6E5   A208     LDX   #$08
F6E7   BD4904   LDA   $0449,X
F6EA   9D0004   STA   $0400,X
F6ED   CA       DEX
F6EE   10F7     BPL   $F6E7
F6F0   20EFF7   JSR   $F7EF
F6F3   204BE7   JSR   $E74B
F6F6   AD3A04   LDA   $043A
F6F9   C92D     CMP   #$2D
F6FB   D007     BNE   $F704
F6FD   A9FE     LDA   #$FE
F6FF   85AE     STA   $AE
F701   4C6FF7   JMP   $F76F
F704   A204     LDX   #$04
F706   BD3404   LDA   $0434,X
F709   95AA     STA   $AA,X
F70B   CA       DEX
F70C   10F8     BPL   $F706
F70E   A200     LDX   #$00
F710   B5AA     LDA   $AA,X
F712   C930     CMP   #$30
F714   D009     BNE   $F71F
F716   A920     LDA   #$20
F718   95AA     STA   $AA,X
F71A   E8       INX
F71B   E004     CPX   #$04
F71D   D0F1     BNE   $F710
F71F   20C9F7   JSR   $F7C9
F722   A202     LDX   #$02
F724   8606     STX   $06
F726   20D5F7   JSR   $F7D5
F729   A606     LDX   $06
F72B   BD31F6   LDA   $F631,X
F72E   209FF7   JSR   $F79F
F731   2086E8   JSR   $E886
F734   20C6F8   JSR   $F8C6
F737   20E2F7   JSR   $F7E2
F73A   A606     LDX   $06
F73C   BD31F6   LDA   $F631,X
F73F   209FF7   JSR   $F79F
F742   20A3E7   JSR   $E7A3
F745   20D5F7   JSR   $F7D5
F748   20EFF7   JSR   $F7EF
F74B   204BE7   JSR   $E74B
F74E   AD3804   LDA   $0438
F751   290F     AND   #$0F
F753   A8       TAY
F754   AD3704   LDA   $0437
F757   290F     AND   #$0F
F759   F009     BEQ   $F764
F75B   AA       TAX
F75C   98       TYA
F75D   18       CLC
F75E   690A     ADC   #$0A
F760   CA       DEX
F761   D0FA     BNE   $F75D
```

F763	A8	TAY	
F764	98	TYA	
F765	2076F7	JSR	$F776
F768	C606	DEC	$06
F76A	10BA	BPL	$F726
F76C	201DF9	JSR	$F91D
F76F	A908	LDA	#$08
F771	85FF	STA	$FF
F773	4CD5CE	JMP	$CED5
F776	48	PHA	
F777	4A	LSR	
F778	A8	TAY	
F779	A506	LDA	$06
F77B	0A	ASL	
F77C	AA	TAX	
F77D	BDDCF5	LDA	$F5DC,X
F780	8500	STA	$00
F782	BDDDF5	LDA	$F5DD,X
F785	8501	STA	$01
F787	B100	LDA	($00),Y
F789	209FF7	JSR	$F79F
F78C	68	PLA	
F78D	2901	AND	#$01
F78F	D008	BNE	$F799
F791	AD1704	LDA	$0417
F794	A606	LDX	$06
F796	95DD	STA	$DD,X
F798	60	RTS	
F799	AD1804	LDA	$0418
F79C	4C94F7	JMP	$F794
F79F	8507	STA	$07
F7A1	A507	LDA	$07
F7A3	290F	AND	#$0F
F7A5	8D1804	STA	$0418
F7A8	A507	LDA	$07
F7AA	4A	LSR	
F7AB	4A	LSR	
F7AC	4A	LSR	
F7AD	4A	LSR	
F7AE	290F	AND	#$0F
F7B0	8D1704	STA	$0417
F7B3	60	RTS	
F7B4	A20A	LDX	#$0A
F7B6	A920	LDA	#$20
F7B8	9D1004	STA	$0410,X
F7BB	CA	DEX	
F7BC	10FA	BPL	$F7B8
F7BE	A20A	LDX	#$0A
F7C0	9D0004	STA	$0400,X
F7C3	CA	DEX	
F7C4	10FA	BPL	$F7C0
F7C6	A20A	LDX	#$0A
F7C8	60	RTS	
F7C9	A20A	LDX	#$0A
F7CB	BD3004	LDA	$0430,X
F7CE	9D2004	STA	$0420,X
F7D1	CA	DEX	
F7D2	10F7	BPL	$F7CB
F7D4	60	RTS	
F7D5	20B4F7	JSR	$F7B4
F7D8	BD2004	LDA	$0420,X
F7DB	9D0004	STA	$0400,X
F7DE	CA	DEX	
F7DF	10F7	BPL	$F7D8
F7E1	60	RTS	
F7E2	20B4F7	JSR	$F7B4
F7E5	BD3004	LDA	$0430,X
F7E8	9D0004	STA	$0400,X
F7EB	CA	DEX	
F7EC	10F7	BPL	$F7E5
F7EE	60	RTS	
F7EF	A20A	LDX	#$0A
F7F1	BD3004	LDA	$0430,X
F7F4	9D1004	STA	$0410,X
F7F7	CA	DEX	
F7F8	10F7	BPL	$F7F1
F7FA	60	RTS	
F7FB	20B4F7	JSR	$F7B4
F7FE	A203	LDX	#$03
F800	BD4004	LDA	$0440,X
F803	9D0504	STA	$0405,X
F806	CA	DEX	
F807	10F7	BPL	$F800
F809	60	RTS	
F80A	20FBF7	JSR	$F7FB
F80D	A965	LDA	#$65
F80F	209FF7	JSR	$F79F
F812	A903	LDA	#$03

```
F814  8D1604  STA  $0416        F884  AD4504  LDA  $0445
F817  20A3E7  JSR  $E7A3        F887  290F    AND  #$0F
F81A  20C9F7  JSR  $F7C9        F889  C903    CMP  #$03
F81D  20FBF7  JSR  $F7FB        F88B  1018    BPL  $F8A5
F820  A904    LDA  #$04         F88D  A940    LDA  #$40
F822  8D1804  STA  $0418        F88F  8500    STA  $00
F825  20B7F8  JSR  $F8B7        F891  A904    LDA  #$04
F828  20C9F7  JSR  $F7C9        F893  8501    STA  $01
F82B  20FBF7  JSR  $F7FB        F895  20DFF8  JSR  $F8DF
F82E  A904    LDA  #$04         F898  F00B    BEQ  $F8A5
F830  8D1604  STA  $0416        F89A  20E2F7  JSR  $F7E2
F833  20B7F8  JSR  $F8B7        F89D  A901    LDA  #$01
F836  20E2F7  JSR  $F7E2        F89F  8D1804  STA  $0418
F839  A203    LDX  #$03         F8A2  204BE7  JSR  $E74B
F83B  BD4004  LDA  $0440,X      F8A5  20E2F7  JSR  $F7E2
F83E  9D1504  STA  $0415,X      F8A8  AD4704  LDA  $0447
F841  CA      DEX               F8AB  8D1704  STA  $0417
F842  10F7    BPL  $F83B        F8AE  AD4804  LDA  $0448
F844  2055EB  JSR  $EB55        F8B1  8D1804  STA  $0418
F847  2055EB  JSR  $EB55        F8B4  4CE9E6  JMP  $E6E9
F84A  204BE7  JSR  $E74B        F8B7  2086E8  JSR  $E886
F84D  20E2F7  JSR  $F7E2        F8BA  20C6F8  JSR  $F8C6
F850  AD4404  LDA  $0444        F8BD  20D5F7  JSR  $F7D5
F853  290F    AND  #$0F         F8C0  20EFF7  JSR  $F7EF
F855  F002    BEQ  $F859        F8C3  4CE9E6  JMP  $E6E9
F857  A90A    LDA  #$0A         F8C6  AD3904  LDA  $0439
F859  18      CLC               F8C9  D001    BNE  $F8CC
F85A  6D4504  ADC  $0445        F8CB  60      RTS
F85D  8507    STA  $07          F8CC  206BEB  JSR  $EB6B
F85F  C607    DEC  $07          F8CF  AD3804  LDA  $0438
F861  A507    LDA  $07          F8D2  C92E    CMP  #$2E
F863  6507    ADC  $07          F8D4  D0F6    BNE  $F8CC
F865  6507    ADC  $07          F8D6  206BEB  JSR  $EB6B
F867  AA      TAX               F8D9  A900    LDA  #$00
F868  BD0DF6  LDA  $F60D,X      F8DB  8D3904  STA  $0439
F86B  8D1604  STA  $0416        F8DE  60      RTS
F86E  BD0EF6  LDA  $F60E,X      F8DF  A003    LDY  #$03
F871  8D1704  STA  $0417        F8E1  B100    LDA  ($00),Y
F874  BD0FF6  LDA  $F60F,X      F8E3  290F    AND  #$0F
F877  8D1804  STA  $0418        F8E5  D00B    BNE  $F8F2
F87A  20E9E6  JSR  $E6E9        F8E7  88      DEY
F87D  AD4404  LDA  $0444        F8E8  B100    LDA  ($00),Y
F880  290F    AND  #$0F         F8EA  290F    AND  #$0F
F882  D021    BNE  $F8A5        F8EC  D004    BNE  $F8F2
```

```
F8EE   A000     LDY   #$00
F8F0   F002     BEQ   $F8F4
F8F2   A002     LDY   #$02
F8F4   B100     LDA   ($00),Y
F8F6   0A       ASL
F8F7   0A       ASL
F8F8   0A       ASL
F8F9   0A       ASL
F8FA   2910     AND   #$10
F8FC   8504     STA   $04
F8FE   C8       INY
F8FF   B100     LDA   ($00),Y
F901   290F     AND   #$0F
F903   0504     ORA   $04
F905   F013     BEQ   $F91A
F907   C904     CMP   #$04
F909   F00F     BEQ   $F91A
F90B   C908     CMP   #$08
F90D   F00B     BEQ   $F91A
F90F   C912     CMP   #$12
F911   F007     BEQ   $F91A
F913   C916     CMP   #$16
F915   F003     BEQ   $F91A
F917   A900     LDA   #$00
F919   60       RTS
F91A   A901     LDA   #$01
F91C   60       RTS
F91D   A202     LDX   #$02
F91F   A000     LDY   #$00
F921   A900     LDA   #$00
F923   85DA     STA   $DA
F925   85DB     STA   $DB
F927   85DC     STA   $DC
F929   B5DD     LDA   $DD,X
F92B   D005     BNE   $F932
F92D   A902     LDA   #$02
F92F   95DA     STA   $DA,X
F931   C8       INY
F932   CA       DEX
F933   10F4     BPL   $F929
F935   98       TYA
F936   D02A     BNE   $F962
F938   A200     LDX   #$00
F93A   A002     LDY   #$02
F93C   B5DD     LDA   $DD,X
F93E   290F     AND   #$0F
F940   9500     STA   $00,X
F942   E8       INX
F943   88       DEY
F944   10F6     BPL   $F93C
F946   2063F9   JSR   $F963
F949   A202     LDX   #$02
F94B   B5DA     LDA   $DA,X
F94D   F010     BEQ   $F95F
F94F   B500     LDA   $00,X
F951   F00C     BEQ   $F95F
F953   2908     AND   #$08
F955   F004     BEQ   $F95B
F957   A903     LDA   #$03
F959   D002     BNE   $F95D
F95B   A901     LDA   #$01
F95D   95DA     STA   $DA,X
F95F   CA       DEX
F960   10E9     BPL   $F94B
F962   60       RTS
F963   A900     LDA   #$00
F965   850E     STA   $0E
F967   A907     LDA   #$07
F969   850F     STA   $0F
F96B   A202     LDX   #$02
F96D   A50F     LDA   $0F
F96F   D001     BNE   $F972
F971   60       RTS
F972   B500     LDA   $00,X
F974   2907     AND   #$07
F976   C50F     CMP   $0F
F978   D00A     BNE   $F984
F97A   B500     LDA   $00,X
F97C   95DA     STA   $DA,X
F97E   850E     STA   $0E
F980   CA       DEX
F981   10EF     BPL   $F972
F983   60       RTS
F984   CA       DEX
F985   10E6     BPL   $F96D
F987   C60F     DEC   $0F
F989   A50E     LDA   $0E
F98B   F0DE     BEQ   $F96B
```

F98D	60	RTS	
F98E	A900	LDA	# $ 00
F990	8581	STA	$ 81
F992	852D	STA	$ 2D
F994	A918	LDA	# $ 18
F996	8582	STA	$ 82
F998	A5AE	LDA	$ AE
F99A	C9FF	CMP	# $ FF
F99C	F007	BEQ	$ F9A5
F99E	C9FE	CMP	# $ FE
F9A0	F008	BEQ	$ F9AA
F9A2	4C13CF	JMP	$ CF13
F9A5	A920	LDA	# $ 20
F9A7	4C9CD2	JMP	$ D29C
F9AA	A91F	LDA	# $ 1F
F9AC	4C9CD2	JMP	$ D29C
F9AF	A900	LDA	# $ 00
F9B1	85DD	STA	$ DD
F9B3	A52D	LDA	$ 2D
F9B5	D024	BNE	$ F9DB
F9B7	A900	LDA	# $ 00
F9B9	85FC	STA	$ FC
F9BB	A681	LDX	$ 81
F9BD	E681	INC	$ 81
F9BF	B5DA	LDA	$ DA,X
F9C1	F013	BEQ	$ F9D6
F9C3	7D70F5	ADC	$ F570,X
F9C6	85C3	STA	$ C3
F9C8	A900	LDA	# $ 00
F9CA	85C6	STA	$ C6
F9CC	A935	LDA	# $ 35
F9CE	852D	STA	$ 2D
F9D0	A903	LDA	# $ 03
F9D2	85FC	STA	$ FC
F9D4	A906	LDA	# $ 06
F9D6	8560	STA	$ 60
F9D8	4C09D7	JMP	$ D709
F9DB	A903	LDA	# $ 03
F9DD	85FC	STA	$ FC
F9DF	4C09D7	JMP	$ D709
F9E2	2013CF	JSR	$ CF13
F9E5	8500	STA	$ 00
F9E7	A000	LDY	# $ 00
F9E9	A202	LDX	# $ 02

F9EB	B5DA	LDA	$ DA,X
F9ED	F006	BEQ	$ F9F5
F9EF	C901	CMP	# $ 01
F9F1	F011	BEQ	$ FA04
F9F3	E600	INC	$ 00
F9F5	CA	DEX	
F9F6	10F3	BPL	$ F9EB
F9F8	A500	LDA	$ 00
F9FA	F00C	BEQ	$ FA08
F9FC	C000	CPY	# $ 00
F9FE	F003	BEQ	$ FA03
FA00	20C2CE	JSR	$ CEC2
FA03	60	RTS	
FA04	C8	INY	
FA05	4CF5F9	JMP	$ F9F5
FA08	A921	LDA	# $ 21
FA0A	4C9CD2	JMP	$ D29C
FA0D	A56B	LDA	$ 6B
FA0F	8587	STA	$ 87
FA11	205DD0	JSR	$ D05D
FA14	F004	BEQ	$ FA1A
FA16	4CF7D0	JMP	$ D0F7
FA19	60	RTS	
FA1A	20C4CF	JSR	$ CFC4
FA1D	A564	LDA	$ 64
FA1F	F0F8	BEQ	$ FA19
FA21	202EFA	JSR	$ FA2E
FA24	F005	BEQ	$ FA2B
FA26	A91D	LDA	# $ 1D
FA28	4C9CD2	JMP	$ D29C
FA2B	4C13CF	JMP	$ CF13
FA2E	20D6FA	JSR	$ FAD6
FA31	A62F	LDX	$ 2F
FA33	A900	LDA	# $ 00
FA35	8500	STA	$ 00
FA37	8501	STA	$ 01
FA39	BD3D06	LDA	$ 063D,X
FA3C	C930	CMP	# $ 30
FA3E	901E	BCC	$ FA5E
FA40	C93A	CMP	# $ 3A
FA42	B01A	BCS	$ FA5E
FA44	A501	LDA	$ 01
FA46	D008	BNE	$ FA50
FA48	8A	TXA	

```
FA49   A400     LDY   $00
FA4B   990200   STA   $0002,Y
FA4E   E600     INC   $00
FA50   E601     INC   $01
FA52   CA       DEX
FA53   10E4     BPL   $FA39
FA55   A500     LDA   $00
FA57   C903     CMP   #$03
FA59   F012     BEQ   $FA6D
FA5B   A9FF     LDA   #$FF
FA5D   60       RTS
FA5E   A501     LDA   $01
FA60   F004     BEQ   $FA66
FA62   A52F     LDA   $2F
FA64   850F     STA   $0F
FA66   A900     LDA   #$00
FA68   8501     STA   $01
FA6A   4C52FA   JMP   $FA52
FA6D   A604     LDX   $04
FA6F   A004     LDY   #$04
FA71   BD3D06   LDA   $063D,X
FA74   C930     CMP   #$30
FA76   9010     BCC   $FA88
FA78   C93A     CMP   #$3A
FA7A   B00C     BCS   $FA88
FA7C   88       DEY
FA7D   30DC     BMI   $FA5B
FA7F   99D000   STA   $00D0,Y
FA82   99DC00   STA   $00DC,Y
FA85   CA       DEX
FA86   10E9     BPL   $FA71
FA88   203DFB   JSR   $FB3D
FA8B   D0CE     BNE   $FA5B
FA8D   20E9FA   JSR   $FAE9
FA90   A603     LDX   $03
FA92   A002     LDY   #$02
FA94   BD3D06   LDA   $063D,X
FA97   C930     CMP   #$30
FA99   9010     BCC   $FAAB
FA9B   C93A     CMP   #$3A
FA9D   B00C     BCS   $FAAB
FA9F   88       DEY
FAA0   30B9     BMI   $FA5B
FAA2   99D400   STA   $00D4,Y
FAA5   99DE00   STA   $00DE,Y
FAA8   CA       DEX
FAA9   10E9     BPL   $FA94
FAAB   20F3FA   JSR   $FAF3
FAAE   D0AB     BNE   $FA5B
FAB0   20E9FA   JSR   $FAE9
FAB3   A602     LDX   $02
FAB5   A002     LDY   #$02
FAB7   BD3D06   LDA   $063D,X
FABA   C930     CMP   #$30
FABC   9010     BCC   $FACE
FABE   C93A     CMP   #$3A
FAC0   B00C     BCS   $FACE
FAC2   88       DEY
FAC3   3096     BMI   $FA5B
FAC5   99D700   STA   $00D7,Y
FAC8   99DE00   STA   $00DE,Y
FACB   CA       DEX
FACC   10E9     BPL   $FAB7
FACE   2007FB   JSR   $FB07
FAD1   D088     BNE   $FA5B
FAD3   A900     LDA   #$00
FAD5   60       RTS
FAD6   A20F     LDX   #$0F
FAD8   A930     LDA   #$30
FADA   95D0     STA   $D0,X
FADC   CA       DEX
FADD   10FB     BPL   $FADA
FADF   A90D     LDA   #$0D
FAE1   85D6     STA   $D6
FAE3   85D9     STA   $D9
FAE5   8D3405   STA   $0534
FAE8   60       RTS
FAE9   A203     LDX   #$03
FAEB   A930     LDA   #$30
FAED   95DC     STA   $DC,X
FAEF   CA       DEX
FAF0   10FB     BPL   $FAED
FAF2   60       RTS
FAF3   A006     LDY   #$06
FAF5   20CEFB   JSR   $FBCE
FAF8   D00C     BNE   $FB06
FAFA   A5D4     LDA   $D4
FAFC   C930     CMP   #$30
```

```
FAFE   D004     BNE   $FB04
FB00   A920     LDA   #$20
FB02   85D4     STA   $D4
FB04   A900     LDA   #$00
FB06   60       RTS
FB07   A5D4     LDA   $D4
FB09   290F     AND   #$0F
FB0B   F002     BEQ   $FB0F
FB0D   A90A     LDA   #$0A
FB0F   18       CLC
FB10   65D5     ADC   $D5
FB12   290F     AND   #$0F
FB14   AA       TAX
FB15   BDCFF5   LDA   $F5CF,X
FB18   C909     CMP   #$09
FB1A   D00E     BNE   $FB2A
FB1C   A9D0     LDA   #$D0
FB1E   8500     STA   $00
FB20   A900     LDA   #$00
FB22   8501     STA   $01
FB24   20DFF8   JSR   $F8DF
FB27   18       CLC
FB28   6909     ADC   #$09
FB2A   A8       TAY
FB2B   20CEFB   JSR   $FBCE
FB2E   D00C     BNE   $FB3C
FB30   A5D7     LDA   $D7
FB32   C930     CMP   #$30
FB34   D004     BNE   $FB3A
FB36   A920     LDA   #$20
FB38   85D7     STA   $D7
FB3A   A900     LDA   #$00
FB3C   60       RTS
FB3D   A000     LDY   #$00
FB3F   C94D     CMP   #$4D
FB41   F00E     BEQ   $FB51
FB43   A001     LDY   #$01
FB45   C954     CMP   #$54
FB47   F008     BEQ   $FB51
FB49   A002     LDY   #$02
FB4B   C953     CMP   #$53
FB4D   F002     BEQ   $FB51
FB4F   A003     LDY   #$03
FB51   20CEFB   JSR   $FBCE
FB54   F001     BEQ   $FB57
FB56   60       RTS
FB57   A60E     LDX   $0E
FB59   E00C     CPX   #$0C
FB5B   F01B     BEQ   $FB78
FB5D   A003     LDY   #$03
FB5F   18       CLC
FB60   B9D000   LDA   $00D0,Y
FB63   7DC1F5   ADC   $F5C1,X
FB66   C93A     CMP   #$3A
FB68   9004     BCC   $FB6E
FB6A   38       SEC
FB6B   E90A     SBC   #$0A
FB6D   38       SEC
FB6E   99D000   STA   $00D0,Y
FB71   99DC00   STA   $00DC,Y
FB74   E8       INX
FB75   88       DEY
FB76   10E8     BPL   $FB60
FB78   A004     LDY   #$04
FB7A   20CEFB   JSR   $FBCE
FB7D   D005     BNE   $FB84
FB7F   A200     LDX   #$00
FB81   4C92FB   JMP   $FB92
FB84   A005     LDY   #$05
FB86   20CEFB   JSR   $FBCE
FB89   D005     BNE   $FB90
FB8B   A201     LDX   #$01
FB8D   4C92FB   JMP   $FB92
FB90   A202     LDX   #$02
FB92   8A       TXA
FB93   48       PHA
FB94   BD7EF5   LDA   $F57E,X
FB97   AA       TAX
FB98   A003     LDY   #$03
FB9A   38       SEC
FB9B   B9D000   LDA   $00D0,Y
FB9E   FDC1F5   SBC   $F5C1,X
FBA1   C930     CMP   #$30
FBA3   B004     BCS   $FBA9
FBA5   18       CLC
FBA6   690A     ADC   #$0A
FBA8   18       CLC
FBA9   993005   STA   $0530,Y
```

```
FBAC    E8        INX
FBAD    88        DEY
FBAE    10EB      BPL    $FB9B
FBB0    68        PLA
FBB1    AA        TAX
FBB2    BDCDF5    LDA    $F5CD,X
FBB5    8D3005    STA    $0530
FBB8    A200      LDX    #$00
FBBA    BD3105    LDA    $0531,X
FBBD    C930      CMP    #$30
FBBF    D00A      BNE    $FBCB
FBC1    A920      LDA    #$20
FBC3    9D3105    STA    $0531,X
FBC6    E8        INX
FBC7    E002      CPX    #$02
FBC9    D0EF      BNE    $FBBA
FBCB    A900      LDA    #$00
FBCD    60        RTS
FBCE    B973F5    LDA    $F573,Y
FBD1    AA        TAX
FBD2    B97EF5    LDA    $F57E,Y
FBD5    850E      STA    $0E
FBD7    A000      LDY    #$00
FBD9    BD89F5    LDA    $F589,X
FBDC    D9DC00    CMP    $00DC,Y
FBDF    F005      BEQ    $FBE6
FBE1    9009      BCC    $FBEC
FBE3    4CFAFB    JMP    $FBFA
FBE6    E8        INX
FBE7    C8        INY
FBE8    C004      CPY    #$04
FBEA    D0ED      BNE    $FBD9
FBEC    A60E      LDX    $0E
FBEE    A000      LDY    #$00
FBF0    BD95F5    LDA    $F595,X
FBF3    D9DC00    CMP    $00DC,Y
FBF6    F005      BEQ    $FBFD
FBF8    B009      BCS    $FC03
FBFA    A9FF      LDA    #$FF
FBFC    60        RTS
FBFD    E8        INX
FBFE    C8        INY
FBFF    C004      CPY    #$04
FC01    D0ED      BNE    $FBF0
FC03    A900      LDA    #$00
FC05    60        RTS
```

发声处理主控程序

```
FC06    A9FF      LDA    #$FF
FC08    8D1740    STA    $4017
FC0B    2098FC    JSR    $FC98
FC0E    A003      LDY    #$03
FC10    B94303    LDA    $0343,Y
FC13    994603    STA    $0346,Y
FC16    A900      LDA    #$00
FC18    994303    STA    $0343,Y
FC1B    88        DEY
FC1C    D0F2      BNE    $FC10
FC1E    AD4903    LDA    $0349
FC21    F00A      BEQ    $FC2D
FC23    A903      LDA    #$03
FC25    8D1540    STA    $4015
FC28    A90F      LDA    #$0F
FC2A    8D1540    STA    $4015
FC2D    A900      LDA    #$00
FC2F    85FF      STA    $FF
FC31    85FE      STA    $FE
FC33    A5FD      LDA    $FD
FC35    8D1140    STA    $4011
FC38    F002      BEQ    $FC3C
FC3A    C6FD      DEC    $FD
FC3C    AD4003    LDA    $0340
FC3F    C920      CMP    #$20
FC41    B005      BCS    $FC48
FC43    A200      LDX    #$00
FC45    8E4103    STX    $0341
FC48    C980      CMP    #$80
FC4A    9003      BCC    $FC4F
FC4C    EE4103    INC    $0341
FC4F    A8        TAY
FC50    4A        LSR
FC51    4A        LSR
FC52    4A        LSR
FC53    4A        LSR
FC54    4A        LSR
FC55    8500      STA    $00
FC57    98        TYA
FC58    AE4103    LDX    $0341
FC5B    D005      BNE    $FC62
```

```
FC5D    38        SEC
FC5E    6500      ADC     $00
FC60    D003      BNE     $FC65
FC62    18        CLC
FC63    E500      SBC     $00
FC65    8D4003    STA     $0340
FC68    60        RTS
```

三个主声部赋值子程序

```
FC69    A07F      LDY     #$7F
FC6B    D005      BNE     $FC72
FC6D    A07F      LDY     #$7F
FC6F    8E0040    STX     $4000
FC72    8C0140    STY     $4001
FC75    A200      LDX     #$00
FC77    A8        TAY
FC78    B90680    LDA     $8006,Y
FC7B    9D0240    STA     $4002,X
FC7E    B90780    LDA     $8007,Y
FC81    0908      ORA     #$08
FC83    9D0340    STA     $4003,X
FC86    60        RTS
FC87    A07F      LDY     #$7F
FC89    8C0540    STY     $4005
FC8C    A204      LDX     #$04
FC8E    D0E7      BNE     $FC77
FC90    A208      LDX     #$08
FC92    C902      CMP     #$02
FC94    F0F0      BEQ     $FC86
FC96    D0DF      BNE     $FC77
```

第一主声部演奏处理

```
FC98    AD4703    LDA     $0347
FC9B    F008      BEQ     $FCA5
FC9D    2069FC    JSR     $FC69
FCA0    A900      LDA     #$00
FCA2    8D4A03    STA     $034A
FCA5    AD4A03    LDA     $034A
FCA8    C936      CMP     #$36
FCAA    B00B      BCS     $FCB7
FCAC    4A        LSR
FCAD    A8        TAY
FCAE    B9E6FC    LDA     $FCE6,Y
FCB1    8D0040    STA     $4000
FCB4    EE4A03    INC     $034A
```

第二主声部演奏处理

```
FCB7    AD4803    LDA     $0348
FCBA    F008      BEQ     $FCC4
FCBC    2087FC    JSR     $FC87
FCBF    A900      LDA     #$00
FCC1    8D4B03    STA     $034B
FCC4    AD4B03    LDA     $034B
FCC7    C936      CMP     #$36
FCC9    B00B      BCS     $FCD6
FCCB    4A        LSR
FCCC    A8        TAY
FCCD    B9E6FC    LDA     $FCE6,Y
FCD0    8D0440    STA     $4004
FCD3    EE4B03    INC     $034B
```

第三主声部演奏处理

```
FCD6    AD4903    LDA     $0349
FCD9    F008      BEQ     $FCE3
FCDB    2090FC    JSR     $FC90
FCDE    A038      LDY     #$38
FCE0    8C0840    STY     $4008
FCE3    4C5BFD    JMP     $FD5B
```

第一主声部音色、音量数据

```
FCE6                                          9F 9D 9C 9B 9A 99 98 97 96 95
FCF0    95 94 94 93 93 93 92 92 92 92 91 91 91 91 91 91
FD00    90 48 4C 4E 50 54 58 5C 60 8C 8C 00 00 00 00 83
FD10    83
```

第一主声部发声处理

```
FD11    A551      LDA     $51
FD13    2907      AND     #$07
FD15    A8        TAY
FD16    BE01FD    LDX     $FD01,Y
FD19    B909FD    LDA     $FD09,Y
FD1C    A8        TAY
FD1D    D008      BNE     $FD27
FD1F    A08C      LDY     #$8C
FD21    A551      LDA     $51
FD23    1002      BPL     $FD27
FD25    A083      LDY     #$83
FD27    8A        TXA
```

```
FD28    60       RTS
FD29    84F0     STY    $F0
FD2B    A95A     LDA    #$5A
FD2D    A003     LDY    #$03
FD2F    84F1     STY    $F1
FD31    A280     LDX    #$80
FD33    206DFC   JSR    $FC6D
FD36    D014     BNE    $FD4C
FD38    84F0     STY    $F0
FD3A    A948     LDA    #$48
FD3C    D0EF     BNE    $FD2D
FD3E    84F0     STY    $F0
FD40    2011FD   JSR    $FD11
FD43    A205     LDX    #$05
FD45    86F1     STX    $F1
FD47    A280     LDX    #$80
FD49    206FFC   JSR    $FC6F
FD4C    C6F1     DEC    $F1
FD4E    D034     BNE    $FD84
FD50    A900     LDA    #$00
FD52    85F0     STA    $F0
FD54    A990     LDA    #$90
FD56    8D0040   STA    $4000
FD59    D029     BNE    $FD84
FD5B    A4FF     LDY    $FF
FD5D    A5F0     LDA    $F0
FD5F    4A       LSR
FD60    46FF     LSR    $FF
FD62    4A       LSR
FD63    B0E7     BCS    $FD4C
FD65    46FF     LSR    $FF
FD67    B0C0     BCS    $FD29
FD69    4A       LSR
FD6A    B0E0     BCS    $FD4C
FD6C    46FF     LSR    $FF
FD6E    B0C8     BCS    $FD38
FD70    4A       LSR
FD71    B0D9     BCS    $FD4C
FD73    46FF     LSR    $FF
FD75    B0C7     BCS    $FD3E
FD77    4A       LSR
FD78    46FF     LSR    $FF
FD7A    4A       LSR
FD7B    46FF     LSR    $FF
FD7D    4A       LSR
FD7E    B017     BCS    $FD97
FD80    46FF     LSR    $FF
FD82    B003     BCS    $FD87
FD84    4C14FE   JMP    $FE14
FD87    84F0     STY    $F0
FD89    A93E     LDA    #$3E
FD8B    85F1     STA    $F1
FD8D    A918     LDA    #$18
FD8F    2075FC   JSR    $FC75
FD92    A986     LDA    #$86
FD94    8D0140   STA    $4001
FD97    A5F1     LDA    $F1
FD99    C938     CMP    #$38
FD9B    D005     BNE    $FDA2
FD9D    A0BD     LDY    #$BD
FD9F    8C0140   STY    $4001
FDA2    4A       LSR
FDA3    B004     BCS    $FDA9
FDA5    A990     LDA    #$90
FDA7    D008     BNE    $FDB1
FDA9    C90A     CMP    #$0A
FDAB    9002     BCC    $FDAF
FDAD    A90A     LDA    #$0A
FDAF    0990     ORA    #$90
FDB1    8D0040   STA    $4000
FDB4    D096     BNE    $FD4C
```

第三主声部发声处理

```
FDB6    A5FD     LDA    $FD
FDB8    18       CLC
FDB9    6903     ADC    #$03
FDBB    C960     CMP    #$60
FDBD    B002     BCS    $FDC1
FDBF    85FD     STA    $FD
FDC1    A908     LDA    #$08
FDC3    8D0840   STA    $4008
FDC6    A5F3     LDA    $F3
FDC8    D007     BNE    $FDD1
FDCA    A97E     LDA    #$7E
FDCC    8D0A40   STA    $400A
FDCF    D004     BNE    $FDD5
FDD1    C6F3     DEC    $F3
FDD3    D01B     BNE    $FDF0
FDD5    A20A     LDX    #$0A
```

```
FDD7  98      TYA
FDD8  2908    AND  #$08
FDDA  F002    BEQ  $FDDE
FDDC  A207    LDX  #$07
FDDE  86F3    STX  $F3
FDE0  98      TYA
FDE1  2940    AND  #$40
FDE3  D026    BNE  $FE0B
FDE5  A551    LDA  $51
FDE7  AA      TAX
FDE8  29F0    AND  #$F0
FDEA  D017    BNE  $FE03
FDEC  A901    LDA  #$01
FDEE  D015    BNE  $FE05
FDF0  98      TYA
FDF1  2908    AND  #$08
FDF3  D01C    BNE  $FE11
FDF5  A5F3    LDA  $F3
FDF7  C905    CMP  #$05
FDF9  D016    BNE  $FE11
FDFB  98      TYA
FDFC  2920    AND  #$20
FDFE  D00B    BNE  $FE0B
FE00  AE4003  LDX  $0340
FE03  A900    LDA  #$00
FE05  8D4303  STA  $0343
FE08  8E0A40  STX  $400A
FE0B  AD4303  LDA  $0343
FE0E  8D0B40  STA  $400B
FE11  4C93FE  JMP  $FE93
    第二主声部发声处理
FE14  A4FC    LDY  $FC
FE16  A5FA    LDA  $FA
FE18  D00A    BNE  $FE24
FE1A  98      TYA
FE1B  2908    AND  #$08
FE1D  D097    BNE  $FDB6
FE1F  98      TYA
FE20  2910    AND  #$10
FE22  D09D    BNE  $FDC1
FE24  98      TYA
FE25  2903    AND  #$03
FE27  D006    BNE  $FE2F
FE29  85FB    STA  $FB
FE2B  85F3    STA  $F3
FE2D  F04B    BEQ  $FE7A
FE2F  45FB    EOR  $FB
FE31  F01E    BEQ  $FE51
FE33  A5FC    LDA  $FC
FE35  85FB    STA  $FB
FE37  0A      ASL
FE38  0A      ASL
FE39  A8      TAY
FE3A  D00D    BNE  $FE49
FE3C  A551    LDA  $51
FE3E  2903    AND  #$03
FE40  85F4    STA  $F4
FE42  A5FC    LDA  $FC
FE44  0A      ASL
FE45  0A      ASL
FE46  05F4    ORA  $F4
FE48  A8      TAY
FE49  B9DCFE  LDA  $FEDC,Y
FE4C  85F4    STA  $F4
FE4E  4C55FE  JMP  $FE55
FE51  C6F6    DEC  $F6
FE53  D025    BNE  $FE7A
FE55  A4F4    LDY  $F4
FE57  E6F4    INC  $F4
FE59  B9ECFE  LDA  $FEEC,Y
FE5C  F0DE    BEQ  $FE3C
FE5E  AA      TAX
FE5F  6A      ROR
FE60  8A      TXA
FE61  2A      ROL
FE62  2A      ROL
FE63  2A      ROL
FE64  2907    AND  #$07
FE66  A8      TAY
FE67  B9F4FF  LDA  $FFF4,Y
FE6A  85F6    STA  $F6
FE6C  8A      TXA
FE6D  293E    AND  #$3E
FE6F  C902    CMP  #$02
FE71  F007    BEQ  $FE7A
FE73  2087FC  JSR  $FC87
FE76  A900    LDA  #$00
FE78  85F8    STA  $F8
```

```
FE7A   A4F8      LDY   $F8
FE7C   C008      CPY   #$08
FE7E   B008      BCS   $FE88
FE80   B98BFE    LDA   $FE8B,Y
FE83   8D0440    STA   $4004
FE86   E6F8      INC   $F8
FE88   4C93FE    JMP   $FE93
```

第二主声部音色音量数据

```
FE8B   94  93  92  92  91
FE90   91  91  90
```

第三主声部发效果音的处理

```
FE93   A5FA      LDA   $FA
FE95   D017      BNE   $FEAE
FE97   A5FE      LDA   $FE
FE99   85FA      STA   $FA
FE9B   F042      BEQ   $FEDF
FE9D   A008      LDY   #$08
FE9F   0A        ASL
FEA0   B003      BCS   $FEA5
FEA2   88        DEY
FEA3   D0FA      BNE   $FE9F
FEA5   B98AFF    LDA   $FF8A,Y
FEA8   85F5      STA   $F5
FEAA   A901      LDA   #$01
FEAC   85F7      STA   $F7
FEAE   C6F7      DEC   $F7
FEB0   D02D      BNE   $FEDF
FEB2   A4F5      LDY   $F5
FEB4   E6F5      INC   $F5
FEB6   B98BFF    LDA   $FF8B,Y
FEB9   D007      BNE   $FEC2
FEBB   85FA      STA   $FA
FEBD   8D0840    STA   $4008
FEC0   F01D      BEQ   $FEDF
FEC2   1009      BPL   $FECD
FEC4   85F9      STA   $F9
FEC6   A4F5      LDY   $F5
FEC8   E6F5      INC   $F5
FECA   B98BFF    LDA   $FF8B,Y
FECD   2090FC    JSR   $FC90
FED0   A5F9      LDA   $F9
FED2   2970      AND   #$70
FED4   4A        LSR
FED5   4A        LSR
FED6   8D0840    STA   $4008
FED9   A5F9      LDA   $F9
FEDB   290F      AND   #$0F
FEDD   85F7      STA   $F7
FEDF   60        RTS
```

发声数据区

FEE0	00	00	0E	22	31	31	45	66	84	84	8D	96	71	6C	AA	B0
FEF0	EC	E6	6A	AA	66	A2	AA	E6	D8	00	70	70	70	6C	AA	B0
FF00	AC	AA	E6	6A	6A	6A	66	A2	AA	67	58	9C	A0	00	DC	A0
FF10	5C	60	E2	A6	62	66	EA	AC	6A	6C	F0	E2	00	B0	AA	A2
FF20	66	6A	AC	A6	A0	62	66	AA	A2	9C	60	62	A6	A0	98	A6
FF30	00	70	6A	70	6A	70	6A	70	6A	6C	66	6C	66	6C	66	6C
FF40	66	6A	62	6A	62	6A	62	6A	62	66	58	5C	60	62	66	6A
FF50	6C	00	70	6C	6A	70	6C	6A	66	6C	6A	66	62	6A	66	58
FF60	5C	60	70	6C	6A	70	6C	6A	66	6C	6A	66	62	6A	E6	00
FF70	3A	34	2C	34	3A	34	2C	34	00	3E	36	2C	36	3E	36	2C
FF80	36	00	3A	30	2A	30	3A	30	2A	30	00	08	11	21	2B	2B
FF90	31	5C	60	B3	30	38	3E	48	50	56	60	00	92	52	60	50
FFA0	5E	4E	5C	4C	5A	4A	58	A4	48	60	78	00	A4	48	56	4C
FFB0	5A	50	5E	52	60	00	A2	48	50	56	60	00	FA	48	C5	48
FFC0	48	FA	48	F5	50	50	50	02	FA	48	FA	50	C5	50	50	FA
FFD0	50	F5	56	56	56	02	FA	50	FA	56	C5	56	56	FA	56	F5
FFE0	60	60	60	60	02	02	00	A4	52	5A	00	92	48	4C	50	54
FFF0	58	5C	60	00	08	0A	14	28	50	1E	ED	00	00	C4	F0	FF
											NMI中断程序入口		复位程序入口			

第八章　BS. 2A 版本的图形库数据介绍

8・1　卡通图形库

卡通图形库占用 4K 字节，每连续的 16 个字节定义一个卡通块，共可定义 256 个卡通块，编号为 0～255。对应的 PPU 地址为＄0000～＄0FFF。卡通图形库数据如下：

0000	00	00	07	1F	3F	1F	0F	30	03	0F	00	02	04	1E	00	03
0010	00	00	E0	F0	F0	F0	C0	40	C0	E0	E0	50	D0	30	00	E0
0020	30	00	04	0F	0F	0C	00	00	0F	1F	27	3F	3F	3C	00	00
0030	48	58	E0	E0	F0	E0	00	00	F0	E0	E0	E0	F0	F8	30	60
0040	00	00	07	1F	3F	1F	0F	00	03	0F	00	02	04	1E	00	03
0050	00	00	E0	F0	F0	F0	C0	80	C0	E0	E0	50	D0	30	00	E0
0060	01	11	1F	0F	07	01	00	00	07	0F	0F	0E	06	07	0E	01
0070	00	80	80	A0	E0	C0	00	00	E0	E0	60	60	60	C0	C0	C0
0080	00	00	00	0F	3F	7F	3F	1F	00	07	1F	01	04	09	3C	00
0090	00	00	00	C0	E0	E0	E0	80	00	80	C0	C0	A0	A0	60	00
00A0	01	02	66	6F	0F	07	00	00	07	0F	1F	1D	0F	07	03	07
00B0	00	00	30	F0	E0	60	00	00	C0	E0	C0	C0	F0	78	08	00
00C0	38	38	31	07	0F	07	03	02	00	03	00	30	31	3F	1C	0F
00D0	00	00	F8	FC	FC	FC	F0	FF	FF	FF	38	94	34	8C	00	F8
00E0	04	04	0F	0F	0F	03	00	00	07	27	3F	3E	3F	03	00	00
00F0	80	86	A6	E0	F0	F0	E0	00	FC	F8	E0	F8	FC	FC	E4	00
0100	00	00	00	01	07	0F	07	03	00	00	03	00	00	01	07	00
0110	00	00	00	F8	FC	FC	FC	F0	00	F0	F8	38	94	34	8C	00
0120	0D	0E	0A	07	03	09	04	00	01	03	07	0F	17	1F	1F	0D
0130	20	40	C0	F2	F6	F0	C2	08	F8	FC	FE	74	F0	F0	C0	00
0140	00	00	00	0C	0F	0F	17	24	03	07	07	07	07	0F	00	1F
0150	00	08	0C	30	F0	F0	E0	20	C0	E0	A0	EC	EC	FC	18	F8
0160	04	06	07	0F	0F	0F	07	00	3F	3F	07	0F	0F	0F	07	0E
0170	20	60	F0	F0	E0	00	00	00	F0	E0	F0	F0	F8	30	00	00
0180	00	00	07	0F	0F	0F	46	C4	03	07	05	01	04	0B	38	38
0190	00	00	E0	F4	F6	F7	62	20	C0	E0	20	40	20	D8	1C	18
01A0	A3	02	07	2F	27	17	08	00	0C	07	35	1F	1F	0F	00	00
01B0	C0	40	E0	E0	F8	F0	64	08	30	E0	A0	E0	F8	FC	78	10
01C0	03	0F	07	0F	07	03	01	18	03	0F	00	02	00	00	00	03
01D0	80	C0	C0	C8	F0	F8	70	20	80	C0	C0	C8	70	F8	F0	E0
01E0	18	03	00	20	10	18	08	00	07	03	07	3F	1F	1F	08	00
01F0	00	C0	40	00	00	00	60	C0	80	80	80	80	C0	E0	60	C0
0200	03	0F	07	0F	07	03	01	00	03	0F	00	02	00	00	00	03

0210	80	C0	C0	C8	F0	F8	70	20	80	C0	C0	C8	70	F8	F0	E0
0220	00	07	01	01	00	00	0C	01	07	03	02	02	07	0F	0F	01
0230	80	00	00	80	00	00	00	C0	C0	C0	80	40	C0	E0	E0	C0
0240	00	07	1F	0F	1F	0F	07	02	00	07	1F	01	05	00	01	01
0250	00	00	80	80	90	E0	F0	E0	00	00	80	80	90	E0	F0	E0
0260	00	00	33	10	00	00	00	0E	07	0F	0B	03	03	07	07	0E
0270	40	00	80	C0	10	30	10	00	C0	C0	50	20	F0	F0	10	00
0280	19	1F	1B	07	03	01	00	00	01	07	00	19	0C	0C	07	03
0290	C0	E0	E0	E4	F8	FC	98	00	C0	E0	60	64	38	7C	78	F0
02A0	00	21	30	10	18	00	00	00	01	21	3F	1F	1F	03	00	00
02B0	08	D8	00	00	00	30	F8	48	D0	C0	C0	D0	E0	F0	18	08
02C0	00	01	07	03	07	03	01	04	00	01	07	00	01	00	00	00
02D0	00	C0	E0	E0	E4	F8	FC	98	00	C0	E0	60	64	38	7C	78
02E0	06	00	01	00	14	16	0A	0A	01	03	01	03	17	1F	0F	0A
02F0	00	08	D8	00	00	02	00	10	F0	D0	C0	C0	E0	F0	C0	00
0300	03	07	07	07	07	03	1E	08	03	07	07	07	07	03	07	17
0310	80	C0	D0	F0	C0	80	80	00	80	C0	C0	C0	D0	B0	E0	C0
0320	00	03	00	00	00	00	06	0C	1F	0B	03	07	0F	0F	06	0C
0330	00	80	00	00	30	00	00	00	80	C0	E0	E0	F0	80	00	00
0340	03	07	07	2F	1F	06	42	E1	03	07	05	29	1C	3C	3C	0E
0350	D0	E4	E6	E7	A0	40	40	80	D0	E0	20	48	3C	38	30	60
0360	20	03	00	00	70	18	00	00	03	03	0F	1F	7F	1B	00	00
0370	00	C0	00	00	00	00	3C	30	C0	C0	E0	F0	F0	F8	3C	30
0380	00	00	20	70	7B	7F	3F	0E	00	00	18	05	03	05	07	07
0390	00	00	02	07	6F	7F	7E	38	00	00	0C	D0	E0	D0	F0	F0
03A0	01	07	17	20	0F	0B	01	00	06	0A	1A	27	03	33	79	00
03B0	C0	F0	F4	02	F0	E8	D0	00	B0	A8	AC	F2	E0	E0	CC	1E
03C0	00	00	00	00	03	07	1F	3E	00	00	18	05	03	05	07	07
03D0	00	00	00	00	60	70	7C	3E	00	00	0C	D0	E0	D0	F0	F0
03E0	79	77	37	20	07	0B	05	00	06	0A	1A	27	03	03	19	3C
03F0	CF	F7	F6	02	F8	E8	C0	00	B0	A8	AC	F2	E0	E6	CF	00
0400	00	00	07	0F	1F	1F	37	B7	00	00	07	0B	11	11	39	79
0410	00	1F	FE	FC	FC	F8	F8	F8	00	00	CE	C0	9C	80	18	38
0420	3F	3F	1F	1F	0F	07	00	00	FB	3F	1F	1F	0F	07	00	00
0430	E0	D8	DC	ED	E2	C0	00	00	F8	E0	80	F3	00	C3	01	00
0440	00	07	0F	1F	1F	37	B7	3F	00	07	0B	11	11	39	79	FF
0450	00	C0	E0	F0	F0	F8	F8	F0	00	C0	E0	F0	F0	F8	38	3B
0460	3F	1F	1F	0F	07	00	00	00	3F	1F	1F	0F	07	00	00	00
0470	ED	EE	A0	F8	80	3C	00	00	13	13	03	81	98	00	00	00
0480	00	00	27	0B	1D	0F	37	3F	00	00	10	3C	1E	1E	0F	03
0490	1C	3E	FF	F0	F0	F0	F8	F8	00	00	C0	60	70	70	F8	F8
04A0	3F	3F	3F	FF	FF	E7	60	20	23	3F	1F	0F	0F	07	00	00
04B0	F8	FC	A1	F2	A0	E0	08	10	F8	FB	A3	F6	A0	D8	38	30
04C0	00	27	0B	1D	0F	37	3F	3F	00	10	3C	1E	1E	0F	03	23

04D0	00	C0	F8	FC	AA	FA	F8	F8	00	C0	60	70	70	F8	F8	F8
04E0	3F	1F	1F	1F	1F	1C	0C	06	3F	1F	1F	0F	07	00	00	00
04F0	FC	FD	AA	E2	E0	60	10	20	FB	F3	F6	E6	C0	10	70	60
0500	00	00	83	E5	6D	3F	3F	1F	01	01	01	03	0B	09	0B	0F
0510	80	00	83	A7	B6	FC	A8	F8	00	80	80	C0	D0	90	D0	F0
0520	1F	1F	0F	0F	0F	07	00	06	1F	1F	0F	0F	07	03	0E	0E
0530	F8	F8	A0	F0	F0	E0	00	60	F8	F8	A0	F0	E0	C0	70	70
0540	00	00	03	05	0D	0F	1F	3F	01	01	01	03	0B	09	1B	1F
0550	80	00	C0	A0	B0	F0	F8	FC	00	80	80	C0	D0	90	D8	F8
0560	7F	FF	EF	CF	82	87	30	1C	1F	1F	4F	4F	47	1B	3C	1C
0570	FE	FF	A3	F3	A3	E1	0C	38	F8	F8	A2	F2	A2	D8	3C	38
0580	00	07	0F	1F	3D	3D	7F	FF	00	01	09	19	2B	0B	59	7F
0590	00	E0	F0	F8	BC	BC	FE	FF	00	80	90	98	DC	DC	9E	FE
05A0	FD	6D	6D	2D	3D	1D	0F	07	78	E0	80	E0	30	18	0F	07
05B0	BF	B6	B6	B4	BC	B8	F0	E0	1E	07	03	07	0C	18	F0	E0
05C0	00	07	0D	1D	3F	3F	7F	FF	00	01	0B	1B	29	09	59	7F
05D0	00	E0	B0	B8	FC	FC	FE	FF	00	80	D0	D8	9C	9C	9E	FE
05E0	FD	6D	6F	3F	3F	1F	0F	07	78	E0	83	DF	3F	1F	0F	07
05F0	BF	B6	F6	FC	FC	F8	F0	E0	1E	07	83	FB	FC	F8	F0	E0
0600	0F	1F	1F	3F	2F	2F	03	01	0F	13	01	00	00	01	7F	FF
0610	00	80	C0	C0	E0	E0	A0	F0	00	80	C0	C0	E0	E0	A0	F0
0620	23	3F	3F	3F	38	1C	07	00	1D	00	00	00	07	03	18	7E
0630	FE	FC	F8	F0	A0	78	20	00	FE	FC	78	70	70	F8	E0	00
0640	01	03	07	0D	0B	03	00	00	01	00	00	00	00	3C	1F	0F
0650	C0	E0	F0	F0	A0	F0	A0	F8	80	E0	70	30	30	70	F0	F8
0660	1D	1F	3F	3F	1F	0F	07	00	03	00	00	00	00	70	38	01
0670	F8	FC	82	FE	A0	F0	F8	10	F8	FC	7E	7E	70	F0	F8	F0
0680	07	0F	0F	1F	1D	1D	1C	30	07	0F	09	18	10	10	13	3F
0690	C0	E0	A0	F0	70	70	7C	1E	C0	E0	20	10	10	10	BC	FE
06A0	38	7F	2B	EF	0F	0F	01	00	3F	78	78	E8	00	00	1E	3F
06B0	3F	F8	A8	F8	F8	E0	80	00	FF	18	08	00	06	1C	78	00
06C0	07	0F	0F	1F	1F	1F	1F	3F	07	0F	0F	1F	1F	1F	1F	3F
06D0	80	E0	A0	F0	80	F0	FC	FF	80	E0	A0	F0	F0	F0	FC	FF
06E0	3F	7F	7F	FF	0F	0F	07	00	3F	7F	2B	EF	0F	0F	1F	3E
06F0	FF	F8	F8	F8	80	F0	C0	80	FF	F8	F8	F8	FE	FC	F8	80
0700	00	00	1E	33	60	45	45	4F	00	00	1E	3F	7F	70	70	70
0710	00	04	00	90	C0	49	48	4C	00	04	00	90	C0	C9	C8	CC
0720	2E	24	60	60	70	3C	1F	07	31	3B	7F	7F	7F	3F	1F	07
0730	DC	B4	6C	08	18	38	30	E0	DC	BC	FC	F8	80	F8	F0	E0
0740	00	00	1F	31	60	55	55	2F	00	00	1F	3F	7F	60	60	30
0750	00	00	00	80	80	C0	40	42	00	00	00	80	80	C0	C0	C2
0760	6F	4F	C6	C0	C0	E0	70	1F	70	70	A9	FF	FF	FF	7F	1F
0770	46	5E	7C	64	08	18	70	C0	C6	DE	FC	FC	F8	F8	F0	C0
0780	00	00	38	38	38	10	03	27	00	00	18	18	18	03	3C	79

0790	00	F8	F8	FE	2E	0E	5E	7E	00	78	78	70	00	F0	A0	F0
07A0	27	27	03	10	38	38	38	00	79	79	3C	03	18	18	18	00
07B0	7E	7E	5E	0E	2E	FE	A8	F8	80	F0	A0	F0	00	70	78	78
07C0	00	00	38	38	38	10	03	27	00	00	30	30	30	03	3C	79
07D0	00	F8	F8	FE	2E	0E	5E	7E	00	98	98	90	00	F0	A0	F0
07E0	27	27	03	10	38	38	38	00	79	79	3C	03	30	30	30	00
07F0	7E	7E	5E	0E	2E	FE	80	F8	A0	F0	A0	F0	00	90	98	98
0800	01	03	01	08	10	07	47	E7	00	01	1D	3E	3F	38	19	6B
0810	00	80	C0	8C	1E	1F	8F	06	00	80	C0	80	06	CF	6F	F2
0820	72	20	00	0C	1E	1F	0F	06	75	27	03	01	06	0E	0F	06
0830	6E	FF	FE	7C	F8	F0	E0	40	D0	E0	40	80	00	00	00	00
0840	01	03	01	08	10	07	47	E7	01	03	1D	3E	3F	38	59	FF
0850	00	80	C0	8C	1E	1F	8F	06	00	80	80	0C	18	D3	67	F2
0860	72	20	00	0C	1E	1F	0F	06	65	07	03	0D	18	12	07	06
0870	6E	FF	FE	7C	F8	F0	E0	40	D0	E0	40	80	00	00	00	00
0880	00	18	1D	1C	18	01	03	03	01	03	1B	1B	1B	02	04	05
0890	00	18	B8	38	18	80	C0	C0	80	C0	D8	D8	D8	40	20	A0
08A0	73	70	7B	79	73	7F	1F	1F	05	77	75	77	75	60	00	00
08B0	CE	0E	DE	9E	CE	FE	F8	F8	A0	EE	AE	EE	AE	06	00	00
08C0	00	18	1D	1C	18	01	03	03	01	1B	1B	1B	03	02	04	05
08D0	00	18	B8	38	18	80	C0	C0	80	D8	D8	D8	C0	40	20	A0
08E0	73	70	7B	79	73	7F	1F	1F	75	07	05	77	75	60	00	00
08F0	CE	0E	DE	9E	CE	FE	F8	F8	AE	E0	A0	EE	AE	06	00	00
0900	07	00	00	01	01	C0	A1	FB	07	07	03	01	01	87	86	9D
0910	E0	00	00	80	80	03	87	FF	E0	E0	C0	80	80	E1	61	B9
0920	DB	81	80	01	01	03	07	07	BD	E6	83	01	01	00	00	07
0930	83	81	01	80	80	C0	A0	E0	BD	67	A3	80	80	00	00	E0
0940	00	10	38	78	B8	1C	09	03	00	10	20	40	C9	7F	0E	0D
0950	00	10	38	3C	3A	70	20	80	00	10	08	04	26	FC	A0	60
0960	07	03	09	7C	F8	60	20	10	1B	0D	0E	1F	89	58	38	10
0970	C0	80	20	7C	3E	0C	08	10	B0	60	E0	F0	22	34	38	10
0980	03	07	07	0F	1F	3F	7F	FF	01	01	01	0F	00	0E	14	1C
0990	E0	F8	FC	FC	FE	80	FE	D4	00	00	00	FC	00	7E	7E	7E
09A0	7F	3D	1E	00	0F	07	01	01	1C	5E	2F	1F	0F	01	07	03
09B0	D4	FE	7E	00	FC	FC	00	00	7E	7E	80	FE	80	00	F8	E0
09C0	00	0F	1F	3F	1F	1F	0F	27	00	01	00	03	25	2E	3C	3C
09D0	F0	F8	FC	FC	FE	FE	DE	AE	00	88	50	20	90	08	24	72
09E0	31	30	19	1C	26	03	01	00	5E	5F	6E	77	3B	1D	0E	03
09F0	76	EA	D4	EC	78	30	90	C0	F8	FC	FE	7E	BC	D8	E0	40
0A00	01	03	03	07	37	37	35	73	02	04	04	0B	1F	5F	5E	7C
0A10	C0	E0	A0	F0	FC	FE	AA	FE	00	00	00	C0	A8	E8	28	0E
0A20	33	37	37	36	37	36	17	06	5C	5B	5B	5B	1B	1B	1B	03
0A30	FE	DE	8A	5E	DC	5C	D8	40	08	E8	A8	E8	A8	E8	E8	E0
0A40	03	07	01	01	03	07	3F	FF	00	00	01	00	03	08	06	1C

0A50	60	70	C0	E0	80	FA	FA	82	80	80	C0	00	F0	1E	0E	7E
0A60	7F	3F	07	03	01	01	07	03	1C	06	08	03	00	01	00	00
0A70	82	FA	AA	F0	E0	C0	70	60	7E	0E	1E	F0	00	C0	80	80
0A80	00	00	38	3C	3F	1F	1F	0F	00	00	00	02	0C	0F	0C	14
0A90	00	70	38	30	F6	FE	F2	70	00	00	00	2C	98	48	20	90
0AA0	0E	0E	4F	7F	7F	27	0C	0E	05	09	04	12	09	1C	10	00
0AB0	38	10	00	84	C8	90	20	00	D8	F8	FC	7C	F8	F0	20	00
0AC0	01	01	03	03	03	47	8B	FF	00	00	00	01	05	03	0B	2A
0AD0	80	80	C0	C0	C0	E2	A3	FF	00	00	00	80	A0	C0	D0	54
0AE0	FF	3E	FE	DE	4E	06	00	07	28	E9	29	09	0D	07	07	07
0AF0	FF	7C	7F	7B	72	60	00	E0	14	97	94	90	B0	E0	E0	E0
0B00	00	00	00	00	01	01	03	3F	00	00	00	08	10	06	0C	00
0B10	00	00	00	00	00	00	80	F8	00	00	00	20	10	C0	60	00
0B20	03	01	01	00	00	00	00	00	0C	06	10	08	00	00	00	00
0B30	80	00	00	00	00	00	00	00	60	C0	10	20	00	00	00	00
0B40	18	B2	22	02	43	33	5F	8F	18	B0	21	15	0C	0C	40	B0
0B50	60	F4	18	18	32	63	81	F2	60	F0	10	A0	CA	93	19	0A
0B60	9F	9F	5B	7F	35	61	89	26	A0	E0	64	40	0A	16	09	26
0B70	B9	E2	F0	DA	A0	84	96	8C	41	1A	0C	24	50	24	16	0C
0B80	08	1C	1C	0C	0C	18	00	00	00	00	20	62	62	A6	FE	FC
0B90	00	00	00	00	00	78	B4	CE	00	00	00	00	00	70	B0	CC
0BA0	01	03	02	03	03	01	00	00	CD	0D	1C	3D	0C	02	0F	1E
0BB0	B6	7A	B4	CE	A3	FE	FC	00	B4	7A	B5	CE	B4	78	02	07
0BC0	08	1C	1C	0C	0C	18	00	01	00	00	20	62	62	A6	FE	FF
0BD0	00	00	00	00	78	B4	CE	B6	00	00	00	00	70	B0	CC	B4
0BE0	03	02	03	03	01	00	00	00	DD	3C	1D	0C	00	03	03	01
0BF0	7A	B4	CE	B7	FE	FC	00	00	7A	B5	CE	B4	78	02	8C	DC
0C00	30	66	D6	D4	F4	E6	82	47	30	60	D0	D2	F2	E0	C2	47
0C10	00	C0	CC	4E	57	DB	9F	DF	00	00	00	8C	96	1A	9F	CF
0C20	65	37	07	37	43	50	54	54	67	3F	0F	7F	4F	57	54	54
0C30	43	FE	A0	F8	82	1A	28	20	83	DE	A0	FC	E2	DA	28	20
0C40	06	06	34	64	D6	D2	00	FF	00	00	32	62	D0	D2	00	FF
0C50	DC	CE	56	5B	8B	8F	83	46	00	0C	94	9A	1E	8F	80	86
0C60	D7	67	17	43	D0	A8	08	00	DF	6F	3F	4F	D7	A8	08	00
0C70	FC	E0	F8	C8	14	54	54	94	DC	E0	F8	EC	D4	54	54	94
0C80	00	00	01	03	0E	06	02	03	00	00	00	00	0C	1C	34	7E
0C90	00	00	80	C0	F0	FD	AB	FF	00	00	00	00	30	3D	3F	FF
0CA0	03	03	03	07	07	00	00	00	FF	0F	7E	1F	07	00	00	00
0CB0	FF	FF	FE	FC	FE	3F	0E	00	CF	FF	E6	7F	B0	3F	0E	00
0CC0	00	00	00	01	03	0E	06	02	00	00	00	00	00	FC	7C	34
0CD0	1E	3F	7F	FE	FC	FD	89	FF	1E	3F	78	7E	30	3D	3D	3F
0CE0	03	03	03	07	0F	03	00	00	1E	0F	1F	3F	7F	03	00	00
0CF0	FF	FF	FE	F0	80	F0	00	00	7F	FF	FE	FF	FA	F0	00	00
0D00	00	00	00	00	18	00	00	00	00	00	00	18	24	18	00	00

0D10	00	00	3C	42	BD	42	3C	00	00	00	3C	7E	83	7E	3C	00
0D20	00	00	00	10	08	00	00	00	00	00	30	28	14	0C	00	00
0D30	00	38	64	5A	5A	26	1C	00	00	78	5C	66	66	3A	1E	00
0D40	00	00	00	10	10	00	00	00	00	00	10	28	28	10	00	00
0D50	10	28	54	54	54	54	28	10	10	38	6C	6C	6C	6C	38	10
0D60	FF	FF	FF	FF	FF	FF	FF	FF	00	00	00	00	00	00	00	00
0D70	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	FF	FF
0D80	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0D90	80	80	80	80	80	80	80	7F	00	00	00	00	00	00	00	00
0DA0	FE	01	01	01	01	01	01	01	00	00	00	00	00	00	00	00
0DB0	FE	FE	82	82	82	FE	FE	00	00	00	00	00	00	00	00	00
0DC0	00	00	00	00	00	00	FE	00	00	00	00	00	00	00	00	00
0DD0	FE	FE	FE	FE	FE	FE	FE	00	00	00	00	00	00	00	00	00
0DE0	3C	24	24	24	24	3C	00	00	3C	3C	3C	3C	3C	3C	00	00
0DF0	00	00	00	10	00	00	00	00	00	00	18	2C	3C	18	00	00
0E00	00	00	00	00	00	00	00	00	C0	60	30	10	18	0C	04	06
0E10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0E20	05	04	0E	0F	0F	07	07	07	02	03	01	00	00	00	00	00
0E30	80	C0	E0	70	78	B8	BC	DC	00	00	00	80	80	60	58	A4
0E40	03	03	03	01	01	00	00	00	00	00	01	01	01	00	00	00
0E50	DC	EE	82	F6	F7	F7	7B	7B	A0	98	14	2A	28	2C	4E	4E
0E60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0E70	3B	1B	1F	0F	0F	07	06	02	CF	1F	1F	0F	0F	07	06	02
0E80	00	00	00	00	00	00	01	03	00	00	00	00	00	00	01	01
0E90	03	0F	1E	3E	76	EE	DC	DC	03	0F	1E	3E	3E	3E	3C	20
0EA0	07	07	0F	0F	1E	1E	1C	3D	01	01	08	08	05	05	03	02
0EB0	BC	B8	78	70	F0	F0	A0	E0	40	40	F8	80	00	00	E0	00
0EC0	39	3B	33	37	27	0E	00	00	06	04	0C	08	18	10	10	30
0ED0	E0	C0	C0	80	00	00	00	00	00	00	00	00	00	00	00	00
0EE0	00	00	00	00	00	00	00	00	20	20	60	40	40	C0	80	80
0EF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0F00	70	70	30	30	30	00	00	00	00	00	00	00	00	00	00	00
0F10	78	18	78	60	78	00	00	00	00	00	00	00	00	00	00	00
0F20	78	18	78	18	78	00	00	00	00	00	00	00	00	00	00	00
0F30	B0	B0	B0	F8	30	00	00	00	00	00	00	00	00	00	00	00
0F40	00	3D	31	31	31	3D	3D	00	FF	FF	FF	FF	FF	FF	FF	FF
0F50	00	F4	B6	B7	B6	F6	F6	00	FF	FF	FF	FF	FF	FF	80	FF
0F60	00	5E	DB	DF	DE	D8	D8	00	FF	FF	FF	FF	FF	FF	FF	FF
0F70	00	6D	6C	6C	6C	7C	7C	00	FF	FF	FF	FF	FF	FF	FF	FF
0F80	00	EF	CC	CF	CC	CF	83	00	FF	FF	FF	FF	FF	FF	FF	FF
0F90	00	78	6C	7C	78	6C	64	00	FF	FF	FF	FF	FF	FF	FF	FF
0FA0	00	3E	36	36	36	3E	3E	00	FF	FF	FF	FF	FF	FF	FF	FF
0FB0	00	F3	8B	FB	A3	C3	80	00	AB	FF	BB	FF	AB	FF	AB	FF
0FC0	00	DE	1B	DF	1E	DB	89	00	AB	FF	FF	FF	FF	FF	FF	FF

0FD0	00	39	6C	6C	7C	7C	6C	00	FF	FF	FF	FF	FF	FF	FF	FF
0FE0	00	EF	CD	CD	81	CF	8F	00	FF	FF	FF	FF	FF	FF	FF	FF
0FF0	00	BC	B6	BE	BC	B6	B2	00	FF	FF	FF	FF	FF	FF	FF	FF

8・2 背景图形库

背景图形库占用4K 字节，每连续的16个字节定义一个背景块，共可定义256个背景块，编号为0～255。对应的 PPU 地址为＄1000～＄1FFF。背景图形库数据如下：

1000	11	FC	00	F8	F8	00	00	00	00	01	03	FB	83	03	FD	FF
1010	7F	BF	80	FF	DF	C0	80	00	00	00	00	1F	1F	00	3F	FF
1020	01	01	01	01	01	01	01	01	FC	FC	FC	FC	FC	FC	A8	FC
1030	40	C0	40	C0	40	C0	40	E0	80	20	A0	00	80	20	B0	18
1040	70	DC	47	C4	44	C4	44	C0	8B	27	BC	3F	BF	3F	BF	3B
1050	40	C0	40	C0	40	C0	40	C0	B8	30	A0	00	80	20	A0	00
1060	00	92	FF	92	92	92	92	00	6D	FF	92	FF	FF	FF	FF	FF
1070	11	11	11	11	11	11	11	11	0C	0C	0C	0C	0C	0C	0C	0C
1080	11	51	A1	51	51	51	51	11	AC	EC	4C	EC	EC	EC	EC	AC
1090	00	00	00	01	07	0C	08	18	00	00	00	00	00	03	07	07
10A0	10	10	10	10	10	10	10	11	0F	0F	0F	0F	0F	0F	0C	0C
10B0	11	10	10	10	10	10	10	10	0C	0C	0F	0F	0F	0F	0F	0F
10C0	18	08	0C	07	01	00	00	00	07	07	03	00	00	00	00	00
10D0	00	00	00	FF	00	00	00	00	00	00	00	00	FF	FF	FF	FF
10E0	00	00	00	00	00	00	00	FF	AB	FF	FF	FF	FF	FF	00	00
10F0	FF	00	00	00	00	00	00	00	00	00	83	FF	FF	FF	FF	FF
1100	00	00	00	00	FF	00	00	00	FF	FF	AB	FF	00	00	00	00
1110	00	00	00	80	E0	30	10	18	00	00	00	00	00	C0	E0	E0
1120	08	08	08	08	08	08	08	88	F0	F0	A0	F0	F0	F0	30	30
1130	88	88	88	88	88	88	88	88	30	30	30	30	30	30	30	30
1140	88	8A	8F	8A	8A	8A	8A	88	35	37	32	37	37	37	37	35
1150	88	08	08	08	08	08	08	08	30	30	F0	F0	A0	F0	F0	F0
1160	80	80	80	80	80	80	80	80	3F	3F	3F	3F	3F	3F	3F	3F
1170	00	49	FF	49	49	49	49	00	B6	FF	49	FF	FF	FF	AB	B6
1180	18	10	30	E0	80	00	00	00	E0	E0	C0	00	00	00	00	00
1190	02	03	02	03	02	03	02	07	01	04	05	00	01	04	0D	18
11A0	0E	3B	A2	23	22	23	22	03	81	E4	3D	FC	A9	FC	A9	DC
11B0	02	03	02	03	02	03	02	03	1D	0C	05	00	01	04	05	00
11C0	00	00	18	3C	3C	18	00	00	FF	C3	99	AD	BD	99	C3	FF
11D0	80	80	80	80	80	80	80	80	00	3F	3F	3F	3F	3F	3F	3F
11E0	00	00	00	00	00	00	00	00	7F	7F	7F	7F	7F	7F	7F	FF
11F0	01	01	01	01	01	01	01	01	00	FC	FC	FC	FC	FC	FC	FC
1200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1210	0C	0C	18	10	00	60	60	00	00	00	00	00	00	00	00	00
1220	28	28	28	00	00	00	00	00	00	00	00	00	00	00	00	00
1230	08	2E	F8	28	3E	E8	20	00	00	00	00	00	00	00	00	00

1240	10	7C	90	FE	12	FC	10	00	00	00	00	00	00	00	00	00
1250	42	A4	A8	54	2A	4A	84	00	00	00	00	00	00	00	00	00
1260	18	28	28	10	6A	44	3A	00	00	00	00	00	00	00	00	00
1270	30	10	20	00	00	00	00	00	00	00	00	00	00	00	00	00
1280	0C	10	20	20	20	10	0C	00	00	00	00	00	00	00	00	00
1290	30	08	04	04	04	08	30	00	00	00	00	00	00	00	00	00
12A0	10	54	38	10	38	54	10	00	00	00	00	00	00	00	00	00
12B0	10	10	10	7C	10	10	10	00	00	00	00	00	00	00	00	00
12C0	00	00	00	00	30	10	20	00	00	00	00	00	00	00	00	00
12D0	00	00	00	7C	00	00	00	00	00	00	00	00	00	00	00	00
12E0	00	00	00	00	00	30	30	00	00	00	00	00	00	00	00	00
12F0	02	04	08	10	20	40	80	00	00	00	00	00	00	00	00	00
1300	7C	C6	8A	92	A2	C6	7C	00	00	00	00	00	00	00	00	00
1310	10	30	10	10	10	10	7C	00	00	00	00	00	00	00	00	00
1320	7C	C6	02	1C	70	C0	FE	00	00	00	00	00	00	00	00	00
1330	FE	04	18	0C	02	C6	7C	00	00	00	00	00	00	00	00	00
1340	1C	34	64	C4	84	FE	04	00	00	00	00	00	00	00	00	00
1350	FC	80	FC	06	02	C6	7C	00	00	00	00	00	00	00	00	00
1360	7C	C0	80	FC	86	C2	7C	00	00	00	00	00	00	00	00	00
1370	FE	86	0C	18	10	10	10	00	00	00	00	00	00	00	00	00
1380	7C	86	C2	7C	86	C2	7C	00	00	00	00	00	00	00	00	00
1390	7C	86	C2	7E	02	06	7C	00	00	00	00	00	00	00	00	00
13A0	00	30	30	00	30	30	00	00	00	00	00	00	00	00	00	00
13B0	00	30	30	00	30	10	20	00	00	00	00	00	00	00	00	00
13C0	08	10	20	40	20	10	08	00	00	00	00	00	00	00	00	00
13D0	00	00	7C	00	7C	00	00	00	00	00	00	00	00	00	00	00
13E0	20	10	08	04	08	10	20	00	00	00	00	00	00	00	00	00
13F0	3C	66	46	0C	08	00	18	00	00	00	00	00	00	00	00	00
1400	7C	82	9A	AA	BC	80	7C	00	00	00	00	00	00	00	00	00
1410	38	6C	C6	82	FE	82	82	00	00	00	00	00	00	00	00	00
1420	FC	86	82	FC	86	82	FC	00	00	00	00	00	00	00	00	00
1430	7C	C6	80	80	80	C6	7C	00	00	00	00	00	00	00	00	00
1440	F8	8C	86	82	86	8C	F8	00	00	00	00	00	00	00	00	00
1450	FE	80	80	FC	80	80	FE	00	00	00	00	00	00	00	00	00
1460	FE	80	80	FC	80	80	80	00	00	00	00	00	00	00	00	00
1470	7C	C6	82	80	8E	C2	7E	00	00	00	00	00	00	00	00	00
1480	82	82	82	FE	82	82	82	00	00	00	00	00	00	00	00	00
1490	7C	10	10	10	10	10	7C	00	00	00	00	00	00	00	00	00
14A0	1E	04	04	04	04	CC	78	00	00	00	00	00	00	00	00	00
14B0	86	8C	98	F0	98	8C	86	00	00	00	00	00	00	00	00	00
14C0	40	40	40	40	40	40	7E	00	00	00	00	00	00	00	00	00
14D0	82	C6	EE	BA	92	82	82	00	00	00	00	00	00	00	00	00
14E0	82	C2	E2	B2	9A	8E	86	00	00	00	00	00	00	00	00	00
14F0	7C	C6	82	82	82	C6	7C	00	00	00	00	00	00	00	00	00

1500	FC	86	82	86	FC	80	80	00	00	00	00	00	00	00	00	00
1510	7C	C6	82	92	8E	C4	7A	00	00	00	00	00	00	00	00	00
1520	FC	86	82	86	F8	8C	86	00	00	00	00	00	00	00	00	00
1530	7C	C0	80	7C	06	02	FC	00	00	00	00	00	00	00	00	00
1540	FE	10	10	10	10	10	10	00	00	00	00	00	00	00	00	00
1550	82	82	82	82	82	C6	7C	00	00	00	00	00	00	00	00	00
1560	82	82	82	C6	6C	38	10	00	00	00	00	00	00	00	00	00
1570	82	82	82	92	BA	EE	C6	00	00	00	00	00	00	00	00	00
1580	82	C6	6C	38	6C	C6	82	00	00	00	00	00	00	00	00	00
1590	82	C6	6C	38	10	10	10	00	00	00	00	00	00	00	00	00
15A0	FE	0C	18	30	60	C0	FE	00	00	00	00	00	00	00	00	00
15B0	00	1C	10	10	10	10	1C	00	00	00	00	00	00	00	00	00
15C0	6C	28	7C	10	7C	10	10	00	00	00	00	00	00	00	00	00
15D0	00	0E	02	02	02	02	0E	00	00	00	00	00	00	00	00	00
15E0	10	28	44	82	00	00	00	00	00	00	00	00	00	00	00	00
15F0	00	00	00	00	00	00	FE	00	00	00	00	00	00	00	00	00
1600	20	38	18	00	00	00	00	00	00	00	00	00	00	00	00	00
1610	00	1C	02	1E	22	1E	00	00	00	00	00	00	00	00	00	00
1620	20	20	20	3C	22	22	3C	00	00	00	00	00	00	00	00	00
1630	00	1E	20	20	20	20	1C	00	00	00	00	00	00	00	00	00
1640	02	02	1E	22	22	22	1E	00	00	00	00	00	00	00	00	00
1650	00	1E	22	22	3E	20	1C	00	00	00	00	00	00	00	00	00
1660	06	08	3E	08	08	08	08	00	00	00	00	00	00	00	00	00
1670	00	1E	22	22	1E	02	1C	00	00	00	00	00	00	00	00	00
1680	20	20	2C	32	22	22	22	00	00	00	00	00	00	00	00	00
1690	08	00	18	08	08	08	1C	00	00	00	00	00	00	00	00	00
16A0	04	00	0C	04	04	04	18	00	00	00	00	00	00	00	00	00
16B0	20	20	24	28	30	28	24	00	00	00	00	00	00	00	00	00
16C0	18	08	08	08	08	08	1C	00	00	00	00	00	00	00	00	00
16D0	00	3C	2A	2A	2A	2A	2A	00	00	00	00	00	00	00	00	00
16E0	00	2C	32	22	22	22	22	00	00	00	00	00	00	00	00	00
16F0	00	1C	22	22	22	22	1C	00	00	00	00	00	00	00	00	00
1700	00	7C	22	22	38	20	78	00	00	00	00	00	00	00	00	00
1710	00	3E	42	42	3E	02	07	00	00	00	00	00	00	00	00	00
1720	00	2C	30	20	20	20	20	00	00	00	00	00	00	00	00	00
1730	00	1E	20	1C	02	02	3C	00	00	00	00	00	00	00	00	00
1740	08	08	3E	08	08	08	06	00	00	00	00	00	00	00	00	00
1750	00	22	22	22	22	26	1A	00	00	00	00	00	00	00	00	00
1760	00	22	22	22	22	14	08	00	00	00	00	00	00	00	00	00
1770	00	22	22	22	2A	2A	14	00	00	00	00	00	00	00	00	00
1780	00	22	14	08	14	22	00	00	00	00	00	00	00	00	00	00
1790	00	22	22	1E	02	02	1C	00	00	00	00	00	00	00	00	00
17A0	00	3E	24	08	10	22	3E	00	00	00	00	00	00	00	00	00
17B0	04	08	08	10	08	08	04	00	00	00	00	00	00	00	00	00

17C0	00	40	20	10	08	04	02	00	00	00	00	00	00	00	00	00
17D0	08	04	04	02	04	04	08	00	00	00	00	00	00	00	00	00
17E0	00	00	00	00	00	42	7E	00	00	00	00	00	00	00	00	00
17F0	00	00	07	18	20	40	80	80	00	00	00	00	00	00	00	00
1800	00	00	00	02	7E	00	00	00	00	00	00	00	00	00	00	00
1810	00	04	3C	00	00	02	7E	00	00	00	00	00	00	00	00	00
1820	00	04	3C	00	18	00	02	7E	00	00	00	00	00	00	00	00
1830	00	7E	A5	A5	81	83	7E	00	00	00	00	00	00	00	00	00
1840	00	FE	20	7C	24	48	FE	00	00	00	00	00	00	00	00	00
1850	08	02	EE	00	14	22	41	00	00	00	00	00	00	00	00	00
1860	20	20	FE	20	20	22	3C	00	00	00	00	00	00	00	00	00
1870	00	28	24	24	42	81	01	00	00	00	00	00	00	00	00	00
1880	20	20	7C	24	24	25	47	00	00	00	00	00	00	00	00	00
1890	10	10	10	FE	10	10	10	00	00	00	00	00	00	00	00	00
18A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
18B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
18C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
18D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
18E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
18F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1900	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1910	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1920	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1930	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1940	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1950	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1960	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1970	4A	40	FC	44	44	44	CC	00	00	00	00	00	00	00	00	00
1980	14	FA	10	FC	10	10	10	00	00	00	00	00	00	00	00	00
1990	04	7A	44	C4	04	0C	38	00	00	00	00	00	00	00	00	00
19A0	4A	40	7E	C8	08	18	30	00	00	00	00	00	00	00	00	00
19B0	04	FA	04	04	04	04	FC	00	00	00	00	00	00	00	00	00
19C0	0A	44	FE	44	04	04	38	00	00	00	00	00	00	00	00	00
19D0	CA	04	C4	0C	08	18	F0	00	00	00	00	00	00	00	00	00
19E0	04	FA	08	18	30	6C	C6	00	00	00	00	00	00	00	00	00
19F0	4A	40	FE	42	44	40	3C	00	00	00	00	00	00	00	00	00
1A00	0A	84	84	4C	08	18	30	00	00	00	00	00	00	00	00	00
1A10	04	7A	44	A4	14	0C	38	00	00	00	00	00	00	00	00	00
1A20	04	7A	10	FE	10	10	30	00	00	00	00	00	00	00	00	00
1A30	0A	A4	A4	04	0C	18	70	00	00	00	00	00	00	00	00	00
1A40	04	7A	00	FE	10	10	30	00	00	00	00	00	00	00	00	00
1A50	4A	40	40	78	4C	40	40	00	00	00	00	00	00	00	00	00
1A60	0A	10	98	88	8C	84	84	00	00	00	00	00	00	00	00	00
1A70	8A	80	FC	80	80	80	7C	00	00	00	00	00	00	00	00	00

1A80	0A	F0	0C	04	0C	18	30	00	00	00	00	00	00	00	00	00
1A90	0A	20	70	D8	8C	06	02	00	00	00	00	00	00	00	00	00
1AA0	0A	10	FE	10	92	92	92	00	00	00	00	00	00	00	00	00
1AB0	16	1A	8E	8C	84	84	84	00	00	00	00	00	00	00	00	00
1AC0	86	8A	FC	80	80	80	7C	00	00	00	00	00	00	00	00	00
1AD0	06	FA	06	04	0C	18	30	00	00	00	00	00	00	00	00	00
1AE0	06	2A	76	D8	8C	06	02	00	00	00	00	00	00	00	00	00
1AF0	16	1A	FE	10	92	92	92	00	00	00	00	00	00	00	00	00
1B00	FF	81	81	81	81	81	81	FF	00	00	00	00	00	00	00	00
1B10	00	00	00	00	70	50	70	00	00	00	00	00	00	00	00	00
1B20	00	00	E0	18	04	02	01	00	00	00	00	00	00	00	00	00
1B30	80	80	40	20	18	07	00	00	00	00	00	00	00	00	00	00
1B40	3C	42	99	A1	A1	99	42	3C	00	00	00	00	00	00	00	00
1B50	00	44	28	10	28	44	00	00	00	00	00	00	00	00	00	00
1B60	10	10	00	7C	00	10	10	00	00	00	00	00	00	00	00	00
1B70	00	01	02	04	18	E0	00	00	00	00	00	00	00	00	00	00
1B80	00	00	00	00	00	00	00	00	00	FE	00	00	00	00	00	00
1B90	00	00	00	00	E7	3C	99	5A	00	00	00	00	00	C3	66	24
1BA0	5A	5A	5A	5A	5A	5A	5A	5A	24	24	24	24	24	24	24	24
1BB0	99	3C	80	00	00	00	00	00	66	C3	00	00	00	00	00	00
1BC0	00	00	00	00	00	00	FF	FF	00	00	00	00	00	FF	00	FF
1BD0	FF	00	AB	00	00	00	00	00	00	FF	00	00	00	00	00	00
1BE0	00	00	00	00	00	00	1F	18	00	00	00	00	00	00	00	00
1BF0	1E	18	18	00	FF	00	FF	FF	00	00	00	00	00	FF	00	FF
1C00	00	7E	3E	1E	0E	06	02	00	FE	82	C6	EE	EE	C6	82	FE
1C10	00	FF	FF	C3	80	C3	81	00	FF	81	83	FF	AB	C3	81	FF
1C20	F7	BB	AA	ED	89	DD	FF	FF	08	44	01	13	B3	BB	BB	FB
1C30	F8	F9	80	00	C7	CF	A3	00	FB	FB	00	00	DF	DF	DF	00
1C40	00	0F	6F	00	00	3F	1F	00	3F	70	90	FF	80	FF	7F	3F
1C50	00	FF	AB	00	00	FF	FF	00	FF	00	00	FF	00	FF	83	FF
1C60	00	F0	F6	00	00	FC	F8	00	FC	0E	09	FF	01	FF	FE	FC
1C70	70	7C	7E	7F	7C	40	40	00	40	40	40	40	40	40	40	E0
1C80	00	00	01	01	02	15	2B	57	01	01	03	17	3E	3D	6F	FF
1C90	80	C0	E0	B0	90	EC	F6	D7	80	C0	E0	F0	B0	EC	F6	FF
1CA0	DD	B9	A3	FD	81	19	33	FF	DD	B9	80	FF	FD	D9	B3	FF
1CB0	B9	DC	82	F3	A1	F9	00	FF	A8	DC	FE	F7	F7	FB	23	FF
1CC0	FF	F1	B9	FF	9F	F0	A0	FF	FF	FF	F9	FF	80	FF	F8	FF
1CD0	00	00	00	20	00	00	00	00	00	00	00	20	02	00	00	00
1CE0	00	00	00	00	00	00	00	80	00	00	00	00	00	00	00	00
1CF0	3C	4E	AF	9F	FF	FF	7E	3C	3C	7E	80	FF	AB	FF	7E	3C
1D00	00	FF	00	00	00	00	00	FF	FF	FF	38	6C	C6	83	FF	FF
1D10	00	FF	00	00	00	00	00	FF	FF	FF	C3	81	81	C3	80	FF
1D20	81	FF	81	81	81	FF	81	81	00	00	00	00	00	00	00	00
1D30	10	28	28	10	10	28	28	10	10	28	28	10	10	28	28	10

1D40	08	08	08	08	28	6C	6E	CA	18	18	18	18	18	18	18	18
1D50	7F	6F	4F	4D	65	7D	7B	5F	80	FF	EF	ED	F5	FD	AB	FF
1D60	00	FF	83	C1	DF	FD	E3	00	FF	FF	FF	E3	DF	FD	80	FF
1D70	00	6C	FE	FE	FE	FE	7C	38	10	7C	FE	9E	BE	FE	7C	38
1D80	0F	3B	77	EF	FF	FF	80	71	00	04	08	10	00	00	00	00
1D90	FF	FF	DD	F3	83	EF	82	B9	00	00	22	0C	00	00	10	39
1DA0	F0	CC	A2	FF	AB	FF	FE	74	00	30	08	00	00	00	00	00
1DB0	3C	7E	5A	BD	80	FF	FF	FF	00	00	24	42	02	00	00	00
1DC0	FF	FF	FF	FF	BF	DB	80	FF	FF	FF	FF	FF	BF	DB	80	FF
1DD0	00	00	00	00	00	00	00	00	24	24	C3	00	00	C3	24	24
1DE0	00	00	00	00	00	00	00	00	00	00	0F	10	20	23	24	24
1DF0	00	00	00	00	00	00	00	00	00	00	F0	08	04	C4	24	24
1E00	00	00	00	00	00	00	00	00	24	24	23	20	10	0F	00	00
1E10	00	00	00	00	00	00	00	00	24	24	C4	04	08	F0	00	00
1E20	00	00	00	00	00	00	00	00	24	24	24	24	24	24	24	24
1E30	00	00	00	00	00	00	00	00	00	00	AB	00	00	FF	00	00
1E40	00	00	00	00	00	00	00	00	00	00	18	24	24	24	24	24
1E50	00	00	00	00	00	00	00	00	24	24	24	24	24	18	00	00
1E60	00	00	00	00	00	00	00	00	00	00	1F	20	20	1F	00	00
1E70	00	00	00	00	00	00	00	00	00	00	F8	04	04	F8	00	00
1E80	18	18	3C	FF	FF	3C	18	18	18	18	3C	FF	FF	3C	18	18
1E90	18	18	3C	FF	00	00	00	00	18	18	3C	FF	FF	00	00	00
1EA0	00	00	00	FF	FF	3C	18	18	00	00	00	FF	AB	3C	18	18
1EB0	18	18	38	F8	F8	38	18	18	18	18	38	F8	F8	38	18	18
1EC0	18	18	1C	1F	1F	1C	18	18	18	18	1C	1F	1F	1C	18	18
1ED0	00	00	00	FF	FF	00	00	00	00	00	00	FF	FF	00	00	00
1EE0	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
1EF0	00	00	00	07	0F	1C	18	18	00	00	00	07	0F	1C	18	18
1F00	00	00	00	E0	F0	38	18	18	00	00	00	E0	F0	38	18	18
1F10	18	18	1C	0F	07	00	00	00	18	18	1C	0F	07	00	00	00
1F20	18	18	38	F0	E0	00	00	00	18	18	38	F0	E0	00	00	00
1F30	FF	C0	A0	90	88	84	82	81	00	3F	5F	6F	77	7B	7D	7E
1F40	81	81	81	81	81	81	81	81	7E	7E	7E	7E	7E	7E	7E	7E
1F50	81	82	84	88	90	A0	C0	FF	7E	7D	7B	77	6F	5F	3F	00
1F60	FF	00	00	00	00	00	00	FF	00	FF	FF	FF	FF	FF	FF	00
1F70	FF	03	05	09	11	21	41	81	00	FC	FA	F6	EE	DE	BE	7E
1F80	81	41	21	11	09	05	03	FF	7E	BE	DE	EE	F6	FA	FC	00
1F90	00	00	00	07	00	07	00	00	00	00	0F	07	0F	07	3F	FF
1FA0	3F	3F	3F	3F	3F	3F	3F	00	40	54	D6	D5	D4	D4	80	FF
1FB0	00	00	30	E0	30	E0	1C	0E	00	00	F0	E0	F0	E0	FC	FF
1FC0	FE	FE	FF	FF	FF	FF	FF	07	02	AA	AB	B3	AB	AB	03	FF
1FD0	FF	FF	FF	FF	FF	FF	FF	FF	00	00	00	00	00	00	00	00
1FE0	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	FF	FF
1FF0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

第九章　6527 指令系统表

符　号 指　令	操　作 过　程	机器码 (16)进制)	N	T	标志寄存器 P NV. BDIZC	寻　址 方　式
ADC # $n	A+ $n+C→A	69 n	2	2	N V . . . . Z C	立即寻址
ADC $n2n1	A+($n2n1)+C→ A	6D n1 n2	3	4	N V . . . . Z C	绝对寻址
ADC $n	A+($n)+C→ A	65 n	2	3	N V . . . . Z C	零页寻址
ADC ($n,X)	A+(ADDRES)+C→ A	61 n	2	6	N V . . . . Z C	先变址间接
ADC ($n),Y	A+(ADDRES)+C→ A	71 n	2	5a	N V . . . . Z C	后变址间接
ADC $n,X	A+($n+X)+C→ A	75 n	2	4	N V . . . . Z C	零页 X 变址
ADC $n2n1,X	A+($n2n1+X)+C→A	7D n1 n2	3	4a	N V . . . . Z C	绝对 X 变址
ADC $n2n1,Y	A+($n2n1+Y)+C→A	79 n1 n2	3	4a	N V . . . . Z C	绝对 Y 变址
AND # $n	A∧$n→ A	29 n	2	2	N . . . . . Z .	立即寻址
AND $n2n1	A∧($n2n1)→ A	2D n1 n2	3	4	N . . . . . Z .	绝对寻址
AND $n	A∧($n)→ A	25 n	2	3	N . . . . . Z .	零页寻址
AND ($n,X)	A∧(ADDRES)→ A	21 n	2	6	N . . . . . Z .	先变址间接
AND ($n),Y	A∧(ADDRES)→ A	31 n	2	5a	N . . . . . Z .	后变址间接
AND $n,X	A∧($n+X)→ A	35 n	2	4	N . . . . . Z .	零页 X 变址
AND $n2n1,X	A∧($n2n1+X)→ A	3D n1 n2	3	4a	N . . . . . Z .	绝对 X 变址
AND $n2n1,Y	A∧($n2n1+Y)→ A	39 n1 n2	3	4a	N . . . . . Z .	绝对 Y 变址
ASL $n2n1		0E n1 n2	3	6	N. . . . . Z C	绝对寻址
ASL $n		06 n	2	5	N . . . . . Z C	零页寻址
ASL	C →D7 →D0 →0 M	0A	1	2	N . . . . . Z C	累加器寻址
ASL $n,X		16 n	2	6	N . . . . . Z C	零页 X 变址
ASL $n2n1,X		1E n1 n2	3	7	N . . . . . Z C	绝对 X 变址
BCC $n	C=0 分支	90 n	2	2b	. . . .	相对寻址
BCS $n	C=1 分支	B0 n	2	2b	. . . .	相对寻址
BEQ $n	Z=1 分支	F0 n	2	2b	. . . .	相对寻址
BNE $n	Z=0 分支	D0 n	2	2b	. . . .	相对寻址
BVC $n	V=0 分支	50 n	2	2b	. . . .	相对寻址
BVS $n	V=1 分支	70 n	2	2b	. . . .	相对寻址
BMI $n	N=1 分支	30 n	2	2b	. . . .	
BPL $n	N=0 分支	10 n	2	2b	. . . .	相对寻址
BIT $n	A∧($n)	24 n	2	3	N V . . . . Z .	零页寻址
BIT $n2n1	A∧($n2n1)	2C n1 n2	3	4	N V . . . . Z .	绝对寻址
BRK	中断	00	1	7	. . . 1 . 1 . .	隐含寻址
CLC	0 →C	18	1	2	 0	隐含寻址
CLD	0 →D	D8	1	2	. . . . 0 . . .	隐含寻址
CLI	0 →I	58	1	2	. . . . . 0 . .	隐含寻址
CLV	0 →V	B8	1	2	. 0	隐含寻址

续表(1)

符号 指令	操作 过程	机器码 (16)进制)	N	T	标志寄存器 P NV. BDIZC	寻址 方式
CMP # $n	A − $n	C9 n	2	2	N . . . . . Z C	立即寻址
CMP $n2n1	A − ($n2n1)	CD n1 n2	3	4	N . . . . . Z C	绝对寻址
CMP ($n,X)	A − (ADDRES)	C1 n	2	6	N . . . . . Z C	先变址间接
CMP ($n),Y	A − (ADDRES)	D1 n	2	5a	N . . . . . Z C	后变址间接
CMP $n,X	A − ($n+X)	D5 n	2	4	N . . . . . Z C	零页 X 变址
CMP $n2n1,X	A − ($n2n1+X)	DD n1 n2	3	4a	N . . . . . Z C	绝对 X 变址
CMP $n2n1,Y	A − ($n2n1+Y)	D9 n1 n2	3	4a	N . . . . . Z C	绝对 Y 变址
CMP $n	A − ($n)	C5 n	2	3	N . . . . . Z C	零页寻址
CPX # $n	X − $n	E0 n	2	2	N . . . . . Z C	立即寻址
CPX $n	X − ($n)	E4 n	2	3	N . . . . . Z C	零页寻址
CPX $n2n1	X − ($n2n1)	EC n1 n2	3	4	N . . . . . Z C	绝对寻址
CPY # $n	Y − $n	C0 n	2	2	N . . . . . Z C	立即寻址
CPY $n	Y − ($n)	C4 n	2	3	N . . . . . Z C	零页寻址
CPY $n2n1	Y − ($n2n1)	CC n1 n2	3	4	N . . . . . Z C	绝对寻址
DEC $n	($n)−1 → ($n)	C6 n	2	5	N . . . . . Z .	零页寻址
DEC $n,X	($n+X)−1→($n+X)	D6 n	2	6	N . . . . . Z .	零页 X 变址
DEC $n2n1	($n2n1)−1→($n2n1)	CE n1 n2	3	6	N . . . . . Z .	绝对寻址
DEC $n2n1,X	($n2n1+X)−1→($n2n1+X)	DE n1 n2	3	7	N . . . . . Z .	绝对 X 变址
DEX	X − 1 →X	CA	1	2	N . . . . . Z .	隐含寻址
DEY	Y − 1 →Y	88	1	2	N . . . . . Z .	隐含寻址
EOR # $n	A⊙$n→ A	49 n	2	2	N . . . . . Z .	立即寻址
EOR $n	A⊙($n)→ A	45 n	2	3	N . . . . . Z .	零页寻址
EOR $n,X	A⊙($n+X) →A	55 n	2	4	N . . . . . Z .	零页 X 变址
EOR $n2n1	A⊙($n2n1) →A	4D n1 n2	3	4	N . . . . . Z .	绝对寻址
EOR $n2n1,X	A⊙($n2n1+X) →A	5D n1 n2	3	4a	N . . . . . Z .	绝对 X 变址
EOR $n2n1,Y	A⊙($n2n1+Y) →A	59 n1 n2	3	4a	N . . . . . Z .	绝对 Y 变址
EOR ($n,X)	A⊙(ADDRES) →A	41 n	2	6	N . . . . . Z .	先变址间接
EOR ($n),Y	A⊙(ADDRES) →A	51 n	2	5a	N . . . . . Z .	后变址间接
INC $n	($n)+1→ ($n)	E6 n	2	5	N . . . . . Z .	零页寻址
INC $n,X	($n+X)+1→($n+X)	F6 n	2	6	N . . . . . Z .	零页 X 变址
INC $n2n1	($n2n1)+1→($n2n1)	EE n1 n2	3	6	N . . . . . Z .	绝对寻址
INC $n2n1,X	($n2n1+X)+1→($n2n1+X)	FE n1 n2	3	7	N . . . . . Z .	绝对 X 变址
INX	X + 1 → X	E8	1	2	N . . . . . Z .	隐含寻址
INY	Y + 1 → Y	C8	1	2	N . . . . . Z .	隐含寻址
JMP $n2n1	$n2n1 →PC	4C n1 n2	3	3	. . . .	绝对寻址
JMP ($n2n1)	($n2n1+1, $n2n1)→PC	6C n1 n2	3	5	. . . .	间接寻址
JSR $n2n1	$n2n1 → PC	20 n1 n2	3	6	. . . .	绝对寻址
LDA # $n	$n →A	A9 n	2	2	N . . . . . Z .	立即寻址
LDA $n	($n) →A	A5 n	2	3	N . . . . . Z .	零页寻址
LDA $n,X	($n+X) →A	B5 n	2	4	N . . . . . Z .	零页 X 变址

续表(2)

符号指令	操作过程	机器码(16)进制)	N	T	标志寄存器P NV. BDIZC	寻址方式
LDA $n2n1	($n2n1)→A	AD n1 n2	3	4	N . . . . . Z .	绝对寻址
LDA $n2n1,X	($n2n1+X)→A	BD n1 n2	3	4a	N . . . . . Z .	绝对X变址
LDA $n2n1,Y	($n2n1+Y)→A	B9 n1 n2	3	4a	N . . . . . Z .	绝对Y变址
LDA ($n,X)	(ADDRES)→A	A1 n	2	6	N . . . . . Z .	先变址间接
LDA ($n),Y	(ADDRES)→A	B1 n	2	5a	N . . . . . Z .	后变址间接
LDX #$n	$n→X	A2 n	2	2	N . . . . . Z .	立即寻址
LDX $n	($n)→X	A6 n	2	3	N . . . . . Z .	零页寻址
LDX $n,Y	($n+Y)→X	B6 n	2	4	N . . . . . Z .	零页Y变址
LDX $n2n1	($n2n1)→X	AE n1 n2	3	4	N . . . . . Z .	绝对寻址
LDX $n2n1,Y	($n2n1+Y)→X	BE n1 n2	3	4a	N . . . . . Z .	绝对Y变址
LDY #$n	$n→Y	A0 n	2	2	N . . . . . Z .	立即寻址
LDY $n	($n)→Y	A4 n	2	3	N . . . . . Z .	零页寻址
LDY $n,X	($n+X)→Y	B4 n	2	4	N . . . . . Z .	零页X变址
LDY $n2n1	($n2n1)→Y	AC n1 n2	3	4	N . . . . . Z .	绝对寻址
LDY $n2n1,X	($n2n1+X)→Y	BC n1 n2	3	4a	N . . . . . Z .	绝对X变址
LSR		4A	1	2	0 . . . . . Z C	累加器寻址
LSR $n		46 n	2	5	0 . . . . . Z C	零页寻址
LSR $n,X	0→D7→D0→C M	56 n	2	6	0 . . . . . Z C	零页X变址
LSR $n2n1		4E n1 n2	3	6	0 . . . . . Z C	绝对寻址
LSR $n2n1,X		5E n1 n2	3	7	0 . . . . . Z C	绝对X变址
NOP		EA	1	2	. . . .	隐含寻址
OR #$n	A∨$n→A	09 n	2	2	N . . . . . Z .	立即寻址
ORA $n	A∨($n)→A	05 n	2	3	N . . . . . Z .	零页寻址
ORA $n,X	A∨($n+X)→A	15 n	2	4	N . . . . . Z .	零页X变址
ORA $n2n1	A∨($n2n1)→A	0D n1 n2	3	4	N . . . . . Z .	绝对寻址
ORA $n2n1,X	A∨($n2n1+X)→A	1D n1 n2	3	4a	N . . . . . Z .	绝对X变址
ORA $n2n1,Y	A∨($n2n1+Y)→A	19 n1 n2	3	4a	N . . . . . Z .	绝对Y变址
ORA ($n,X)	A∨(ADDRES)→A	01 n	2	6	N . . . . . Z .	先变址间接
ORA ($n),Y	A∨(ADDRES)→A	11 n	2	5a	N . . . . . Z .	后变址间接
PHA	A→Ms,S−1→S	48	1	3	. . . .	隐含寻址
PHP	P→Ms,S−1→S	08	1	3	. . . .	隐含寻址
PLA	S+1→S,Ms→A	68	1	4	N . . . . . Z .	隐含寻址
PLP	S+1→S,Ms→P	28	1	4	恢复P原来的状态	隐含寻址
ROL		2A	1	2	N . . . . . Z C	累加器寻址
ROL $n		26 n	2	5	N . . . . . Z C	零页寻址
ROL $n,X	←D7←D0←C← M	36 n	2	6	N . . . . . Z C	零页X变址
ROL $n2n1		2E n1 n2	3	6	N . . . . . Z C	绝对寻址
ROL $n2n1,X		3E n1 n2	3	7	N . . . . . Z C	绝对X变址

续表(3)

符　号 指　令	操　作 过　程	机器码 (16)进制)	N	T	标志寄存器 P NV.BDIZC	寻　址 方　式
ROR		6A	1	2	N . . . . . Z C	累加器寻址
ROR $n		66 n	2	5	N . . . . . Z C	零页寻址
ROR $n,X	→C→D7→D0→	76 n	2	6	N . . . . . Z C	零页 X 变址
ROR $n2n1	M	6E n1 n2	3	6	N . . . . . Z C	绝对寻址
ROR $n2n1,X		7E n1 n2	3	7	N . . . . . Z C	绝对 X 变址
RTI	中断返回	40	1	6	恢复 P 原来的状态	隐含寻址
RTS	子程序返回	60	1	6	. . . .	隐含寻址
SBC # $n	A－$n－C →A	E9 n	2	2	N V C	立即寻址
SBC $n	A－($n)－C →A	E5 n	2	3	N V C	零页寻址
SBC $n,X	A－($n+X)－C →A	F5 n	2	4	N V C	零页 X 变址
SBC $n2n1	A－($n2n1)－C →A	ED n1 n2	3	4	N V C	绝对寻址
SBC $n2n1,X	A－($n2n1+X)－C→A	FD n1 n2	3	4a	N V . . . . Z C	绝对 X 变址
SBC $n2n1,Y	A－($n2n1+Y)－C→A	F9 n1 n2	3	4a	N V . . . . Z C	绝对 Y 变址
SBC ($n,X)	A－(ADDRES)－C →A	E1 n	2	6	N V . . . . Z C	先变址间接
SBC ($n),Y	A－(ADDRES)－C →A	F1 n	2	5a	N V . . . . Z C	后变址间接
SEC	1 →C	38	1	2	 1	隐含寻址
SED	1 →D	F8	1	2	. . . . 1 . . .	隐含寻址
SEI	1 →I	78	1	2	. . . . . 1 . .	隐含寻址
STA $n	A →($n)	85 n	2	3	. . . .	零页寻址
STA $n,X	A →($n+X)	95 n	2	4	. . . .	零页 X 变址
STA $n2n1	A →($n2n1)	8D n1 n2	3	4	. . . .	绝对寻址
STA $n2n1,X	A →($n2n1+X)	9D n1 n2	3	5	. . . .	绝对 X 变址
STA $n2n1,Y	A →($n2n1+Y)	99 n1 n2	3	5	. . . .	绝对 Y 变址
STA ($n,X)	A →(ADDRES)	81 n	2	6	. . . .	先变址间接
STA ($n),Y	A →(ADDRES)	91 n	2	6	. . . .	后变址间接
STX $n	X →($n)	86 n	2	3	. . . .	零页寻址
STX $n,Y	X →($n+Y)	96 n	2	4	. . . .	零页 Y 变址
STX $n2n1	X →($n2n1)	8E n1 n2	3	4	. . . .	绝对寻址
STY $n	Y →($n)	84 n	2	3	. . . .	零页寻址
STY $n,X	Y →($n+X)	94 n	2	4	. . . .	零页 X 变址
STY $n2n1	Y →($n2n1)	8C n1 n2	3	4	. . . .	绝对寻址
TAX	A →X	AA	1	2	N . . . . . Z .	隐含寻址
TAY	A →Y	A8	1	2	N . . . . . Z .	隐含寻址
TSX	S →X	BA	1	2	N . . . . . Z .	隐含寻址
TXA	X →A	8A	1	2	N . . . . . Z .	隐含寻址
TXS	X →S	9A	1	2	. . . .	隐含寻址
TYA	Y →A	98	1	2	N . . . . . Z .	隐含寻址

说明：1. 表中的符号

N——字节数。

T——时钟周期数。

n ——代表两位十六进制数。

n1n2——代表四位十六进制数。

ADDRES——代表有效地址。

⊙——表示异或运算

2. 表中的注释

a——如果跨页，指令执行时再加一个时钟周期。

b——不分支时 T=2，分支时 T=3；如果跨页再加一个时钟周期。

附录一　背景显示单元与PPU地址对照表

附表1　背景00页显示单元与PPU地址对照表

行　次	PPU　地　址	行　次	PPU　地　址
00	$2000～$201F	10	$2200～$221F
01	$2020～$203F	11	$2220～$223F
02	$2040～$205F	12	$2240～$225F
03	$2060～$207F	13	$2260～$226F
04	$2080～$209F	14	$2280～$229F
05	$20A0～$20BF	15	$22A0～$22BF
06	$20C0～$20DF	16	$22C0～$22DF
07	$20E0～$20FF	17	$22E0～$22FF
08	$2100～$211F	18	$2300～$231F
09	$2120～$213F	19	$2320～$233F
0A	$2140～$215F	1A	$2340～$235F
0B	$2160～$217F	1B	$2360～$237F
0C	$2180～$219F	1C	$2380～$239F
0D	$21A0～$21BF	1D	$23A0～$23BF
0E	$21C0～$21DF		
0F	$21E0～$21FF		

附表 2　背景 10 页显示单元与 PPU 地址对照表

行　次	PPU　地　址	行　次	PPU　地　址
00	$2400～$241F	10	$2600～$261F
01	$2420～$243F	11	$2620～$263F
02	$2440～$245F	12	$2640～$265F
03	$2460～$247F	13	$2660～$266F
04	$2480～$249F	14	$2680～$269F
05	$24A0～$24BF	15	$26A0～$26BF
06	$24C0～$24DF	16	$26C0～$26DF
07	$24E0～$24FF	17	$26E0～$26FF
08	$2500～$251F	18	$2700～$271F
09	$2520～$253F	19	$2720～$273F
0A	$2540～$255F	1A	$2740～$275F
0B	$2560～$257F	1B	$2760～$277F
0C	$2580～$259F	1C	$2780～$279F
0D	$25A0～$25BF	1D	$27A0～$27BF
0E	$25C0～$25DF		
0F	$25E0～$25FF		

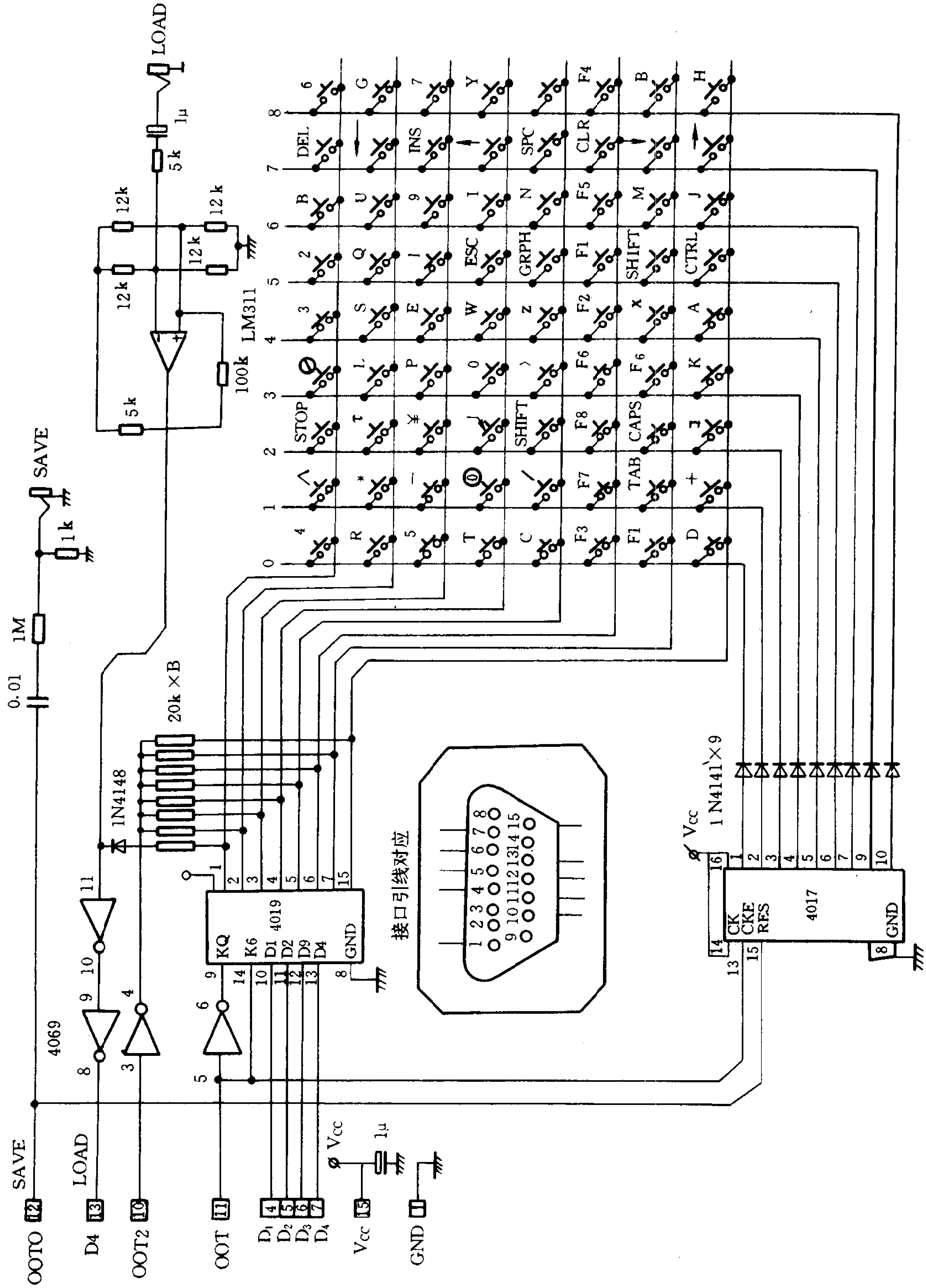
附表 3　背景 01 页显示单元与 PPU 地址对照表

行　次	PPU 地址	行　次	PPU 地址
00	$ 2800～$ 281F	10	$ 2A00～$ 2A1F
01	$ 2820～$ 283F	11	$ 2A20～$ 2A3F
02	$ 2840～$ 285F	12	$ 2A40～$ 2A5F
03	$ 2860～$ 287F	13	$ 2A60～$ 2A6F
04	$ 2880～$ 289F	14	$ 2A80～$ 2A9F
05	$ 28A0～$ 28BF	15	$ 2AA0～$ 2ABF
06	$ 28C0～$ 28DF	16	$ 2AC0～$ 2ADF
07	$ 28E0～$ 28FF	17	$ 2AE0～$ 2AFF
08	$ 2900～$ 291F	18	$ 2B00～$ 2B1F
09	$ 2920～$ 293F	19	$ 2B20～$ 2B3F
0A	$ 2940～$ 295F	1A	$ 2B40～$ 2B5F
0B	$ 2960～$ 297F	1B	$ 2B60～$ 2B7F
0C	$ 2980～$ 299F	1C	$ 2B80～$ 2B9F
0D	$ 29A0～$ 29BF	1D	$ 2BA0～$ 2BBF
0E	$ 29C0～$ 29DF		
0F	$ 29E0～$ 29FF		

附表4　背景11页显示单元与PPU地址对照表

行　次	PPU　地　址	行　次	PPU　地　址
00	$ 2C00～$ 2C1F	10	$ 2E00～$ 2E1F
01	$ 2C20～$ 2C3F	11	$ 2E20～$ 2E3F
02	$ 2C40～$ 2C5F	12	$ 2E40～$ 2E5F
03	$ 2C60～$ 2C7F	13	$ 2E60～$ 2E6F
04	$ 2C80～$ 2C9F	14	$ 2E80～$ 2E9F
05	$ 2CA0～$ 2CBF	15	$ 2EA0～$ 2EBF
06	$ 2CC0～$ 2CDF	16	$ 2EC0～$ 2EDF
07	$ 2CE0～$ 2CFF	17	$ 2EE0～$ 2EFF
08	$ 2D00～$ 2D1F	18	$ 2F00～$ 2F1F
09	$ 2D20～$ 2D3F	19	$ 2F20～$ 2F3F
0A	$ 2D40～$ 2D5F	1A	$ 2F40～$ 2F5F
0B	$ 2D60～$ 2D7F	1B	$ 2F60～$ 2F7F
0C	$ 2D80～$ 2D9F	1C	$ 2F80～$ 2F9F
0D	$ 2DA0～$ 2DBF	1D	$ 2FA0～$ 2FBF
0E	$ 2DC0～$ 2DDF		
0F	$ 2DE0～$ 2DFF		

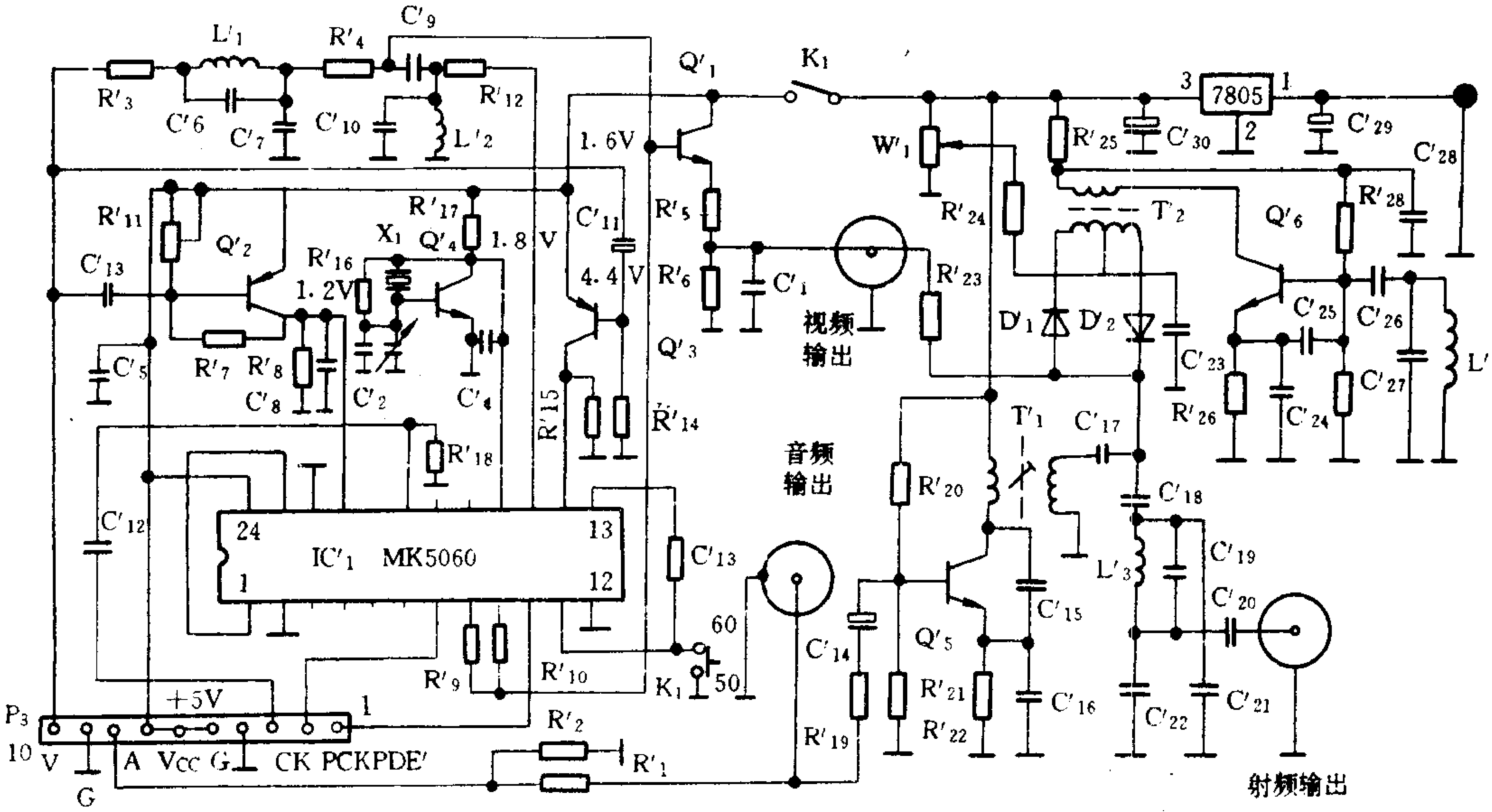
附录二　键盘电路图



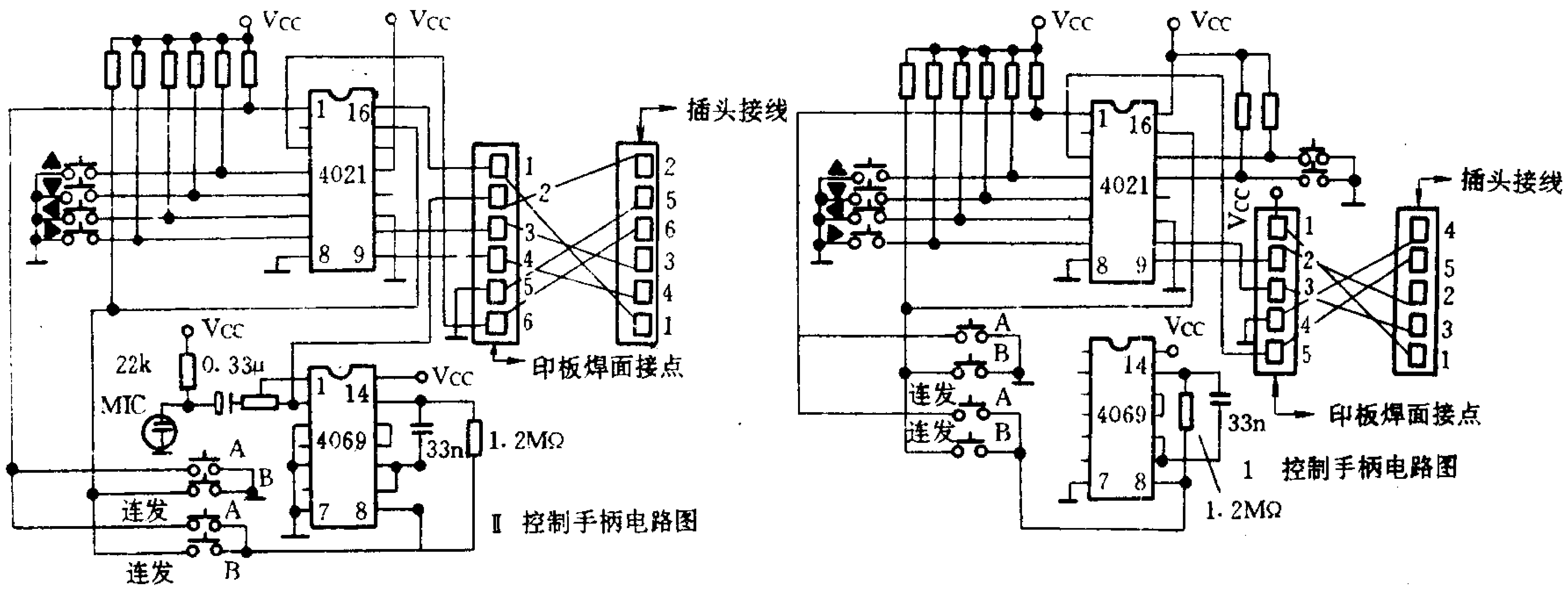
裕兴键盘电路原理图

附示三　家用电视游戏机电路图选集

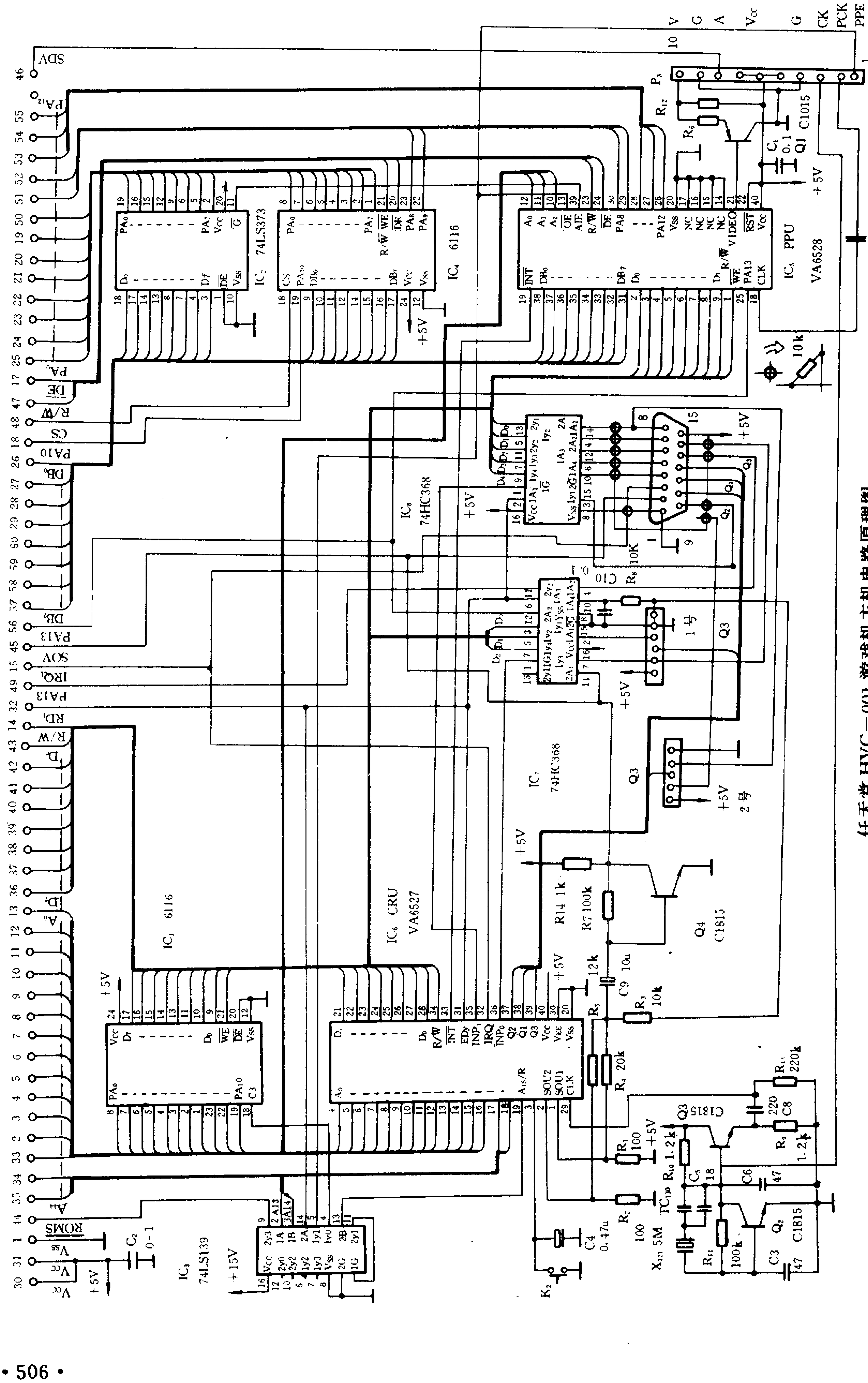
1. 任天堂 HVC—001 电路原理图



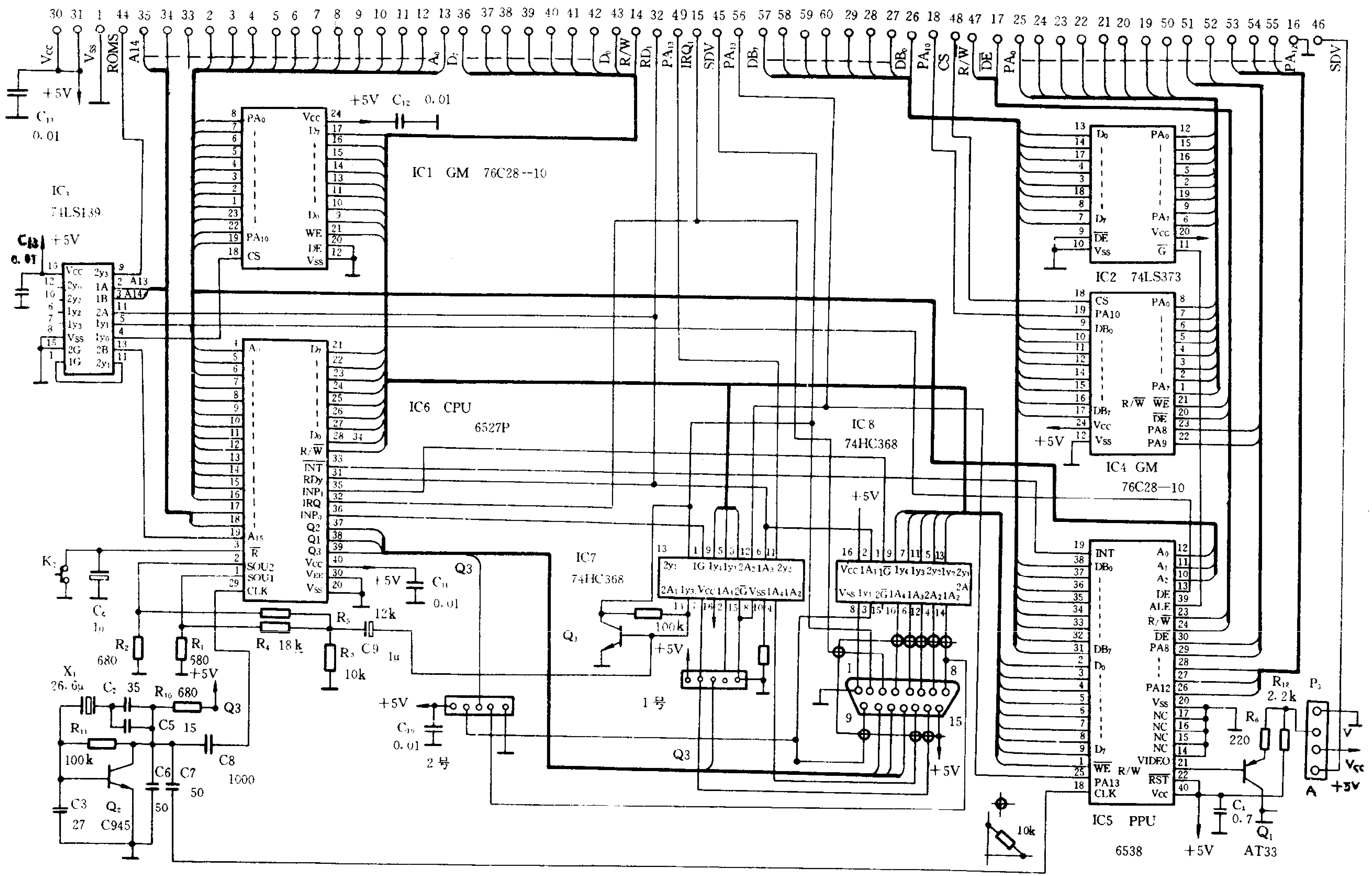
HVC—001 调制板



HVC--001 控制手柄电路

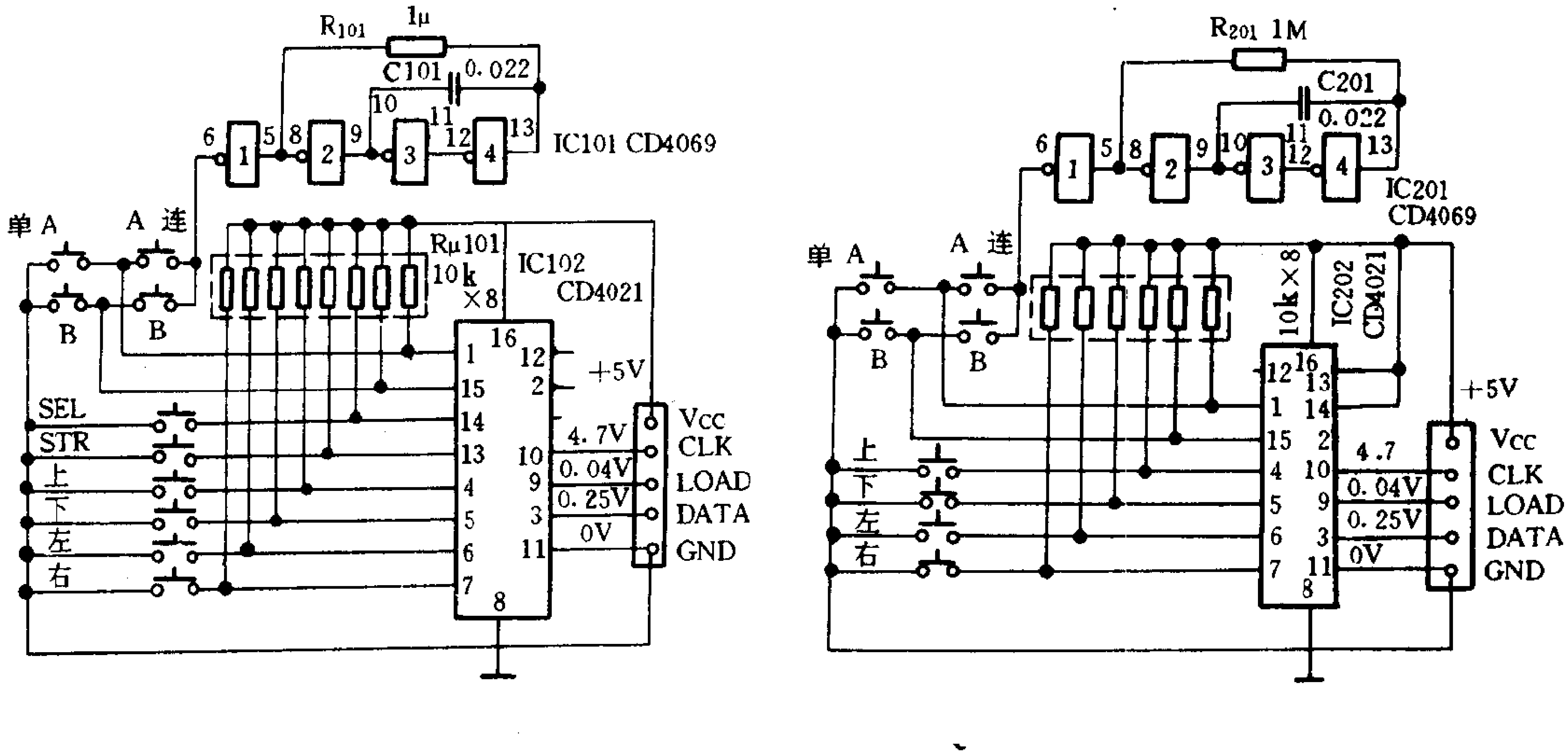


任天堂 HVC—001 游戏机主机电路原理图

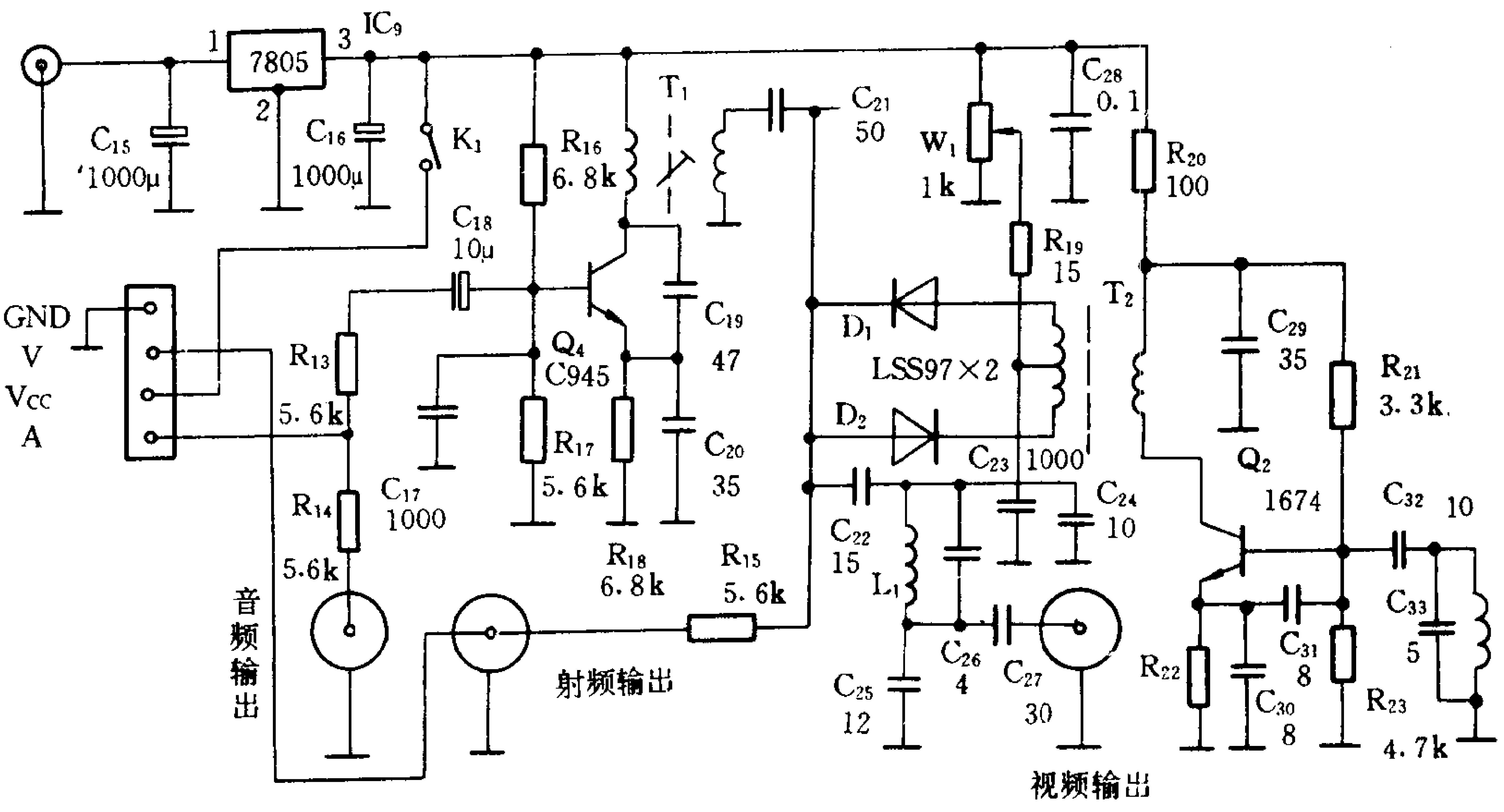


任天堂 FH989 游戏机主机电路原理图

2. 任天堂 FH989 电路原理图

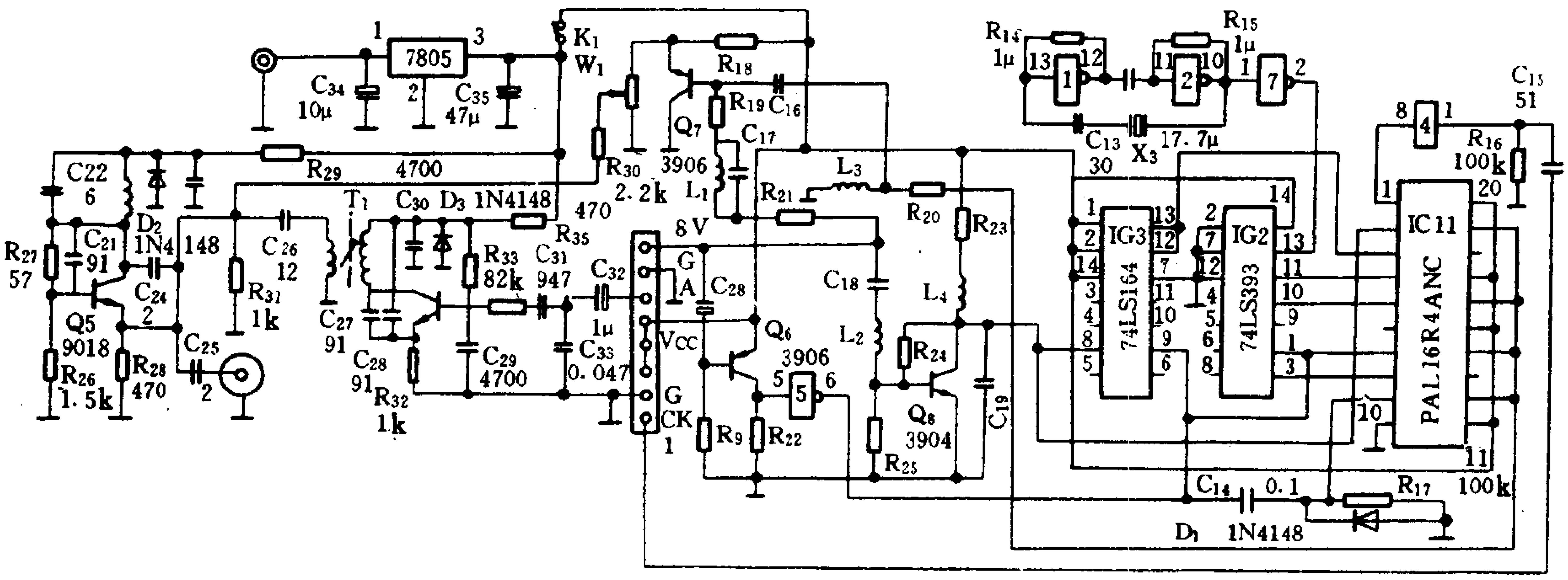


FH989 调制板

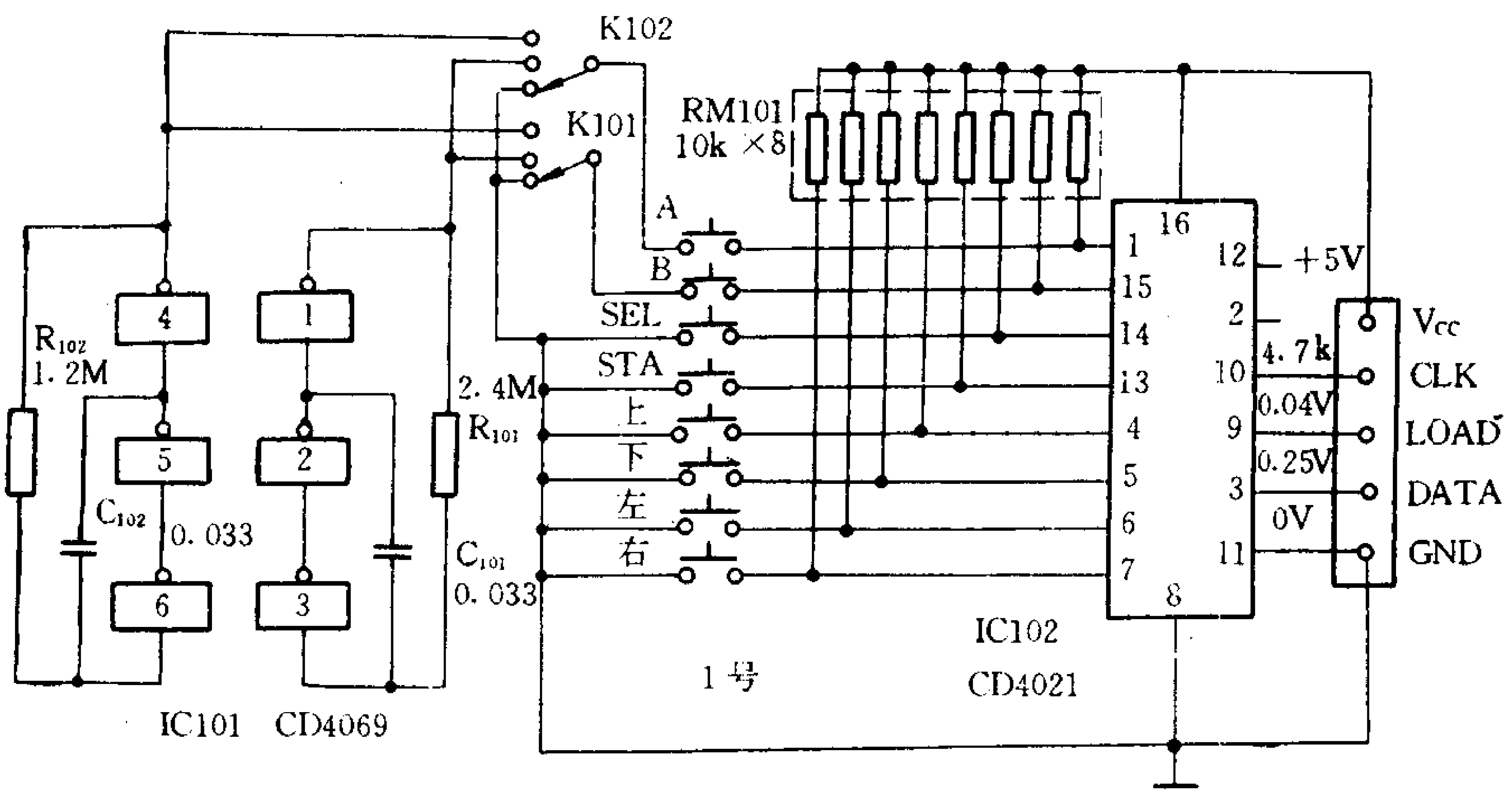


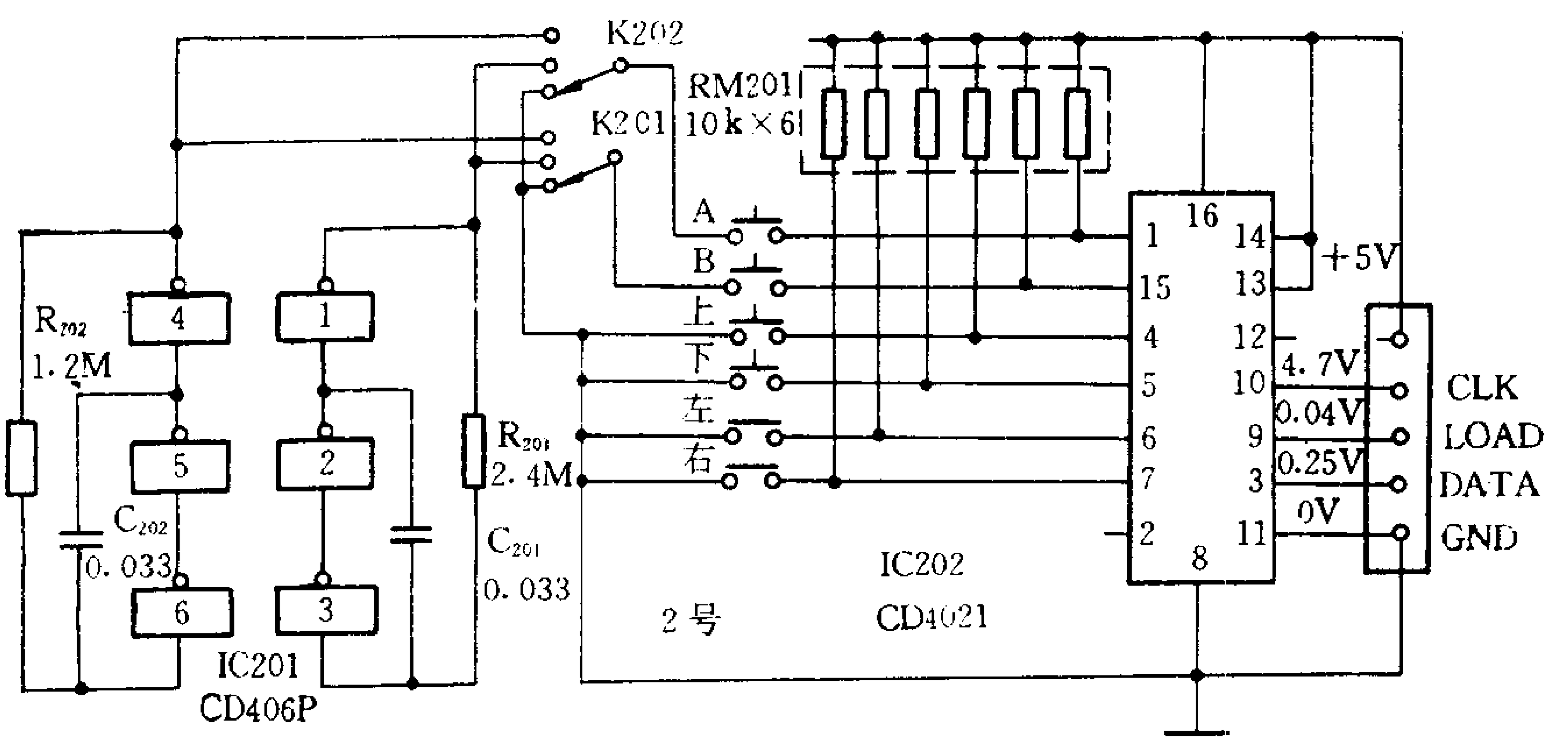
FH989 手柄

3. 任天堂 180 电路原理图

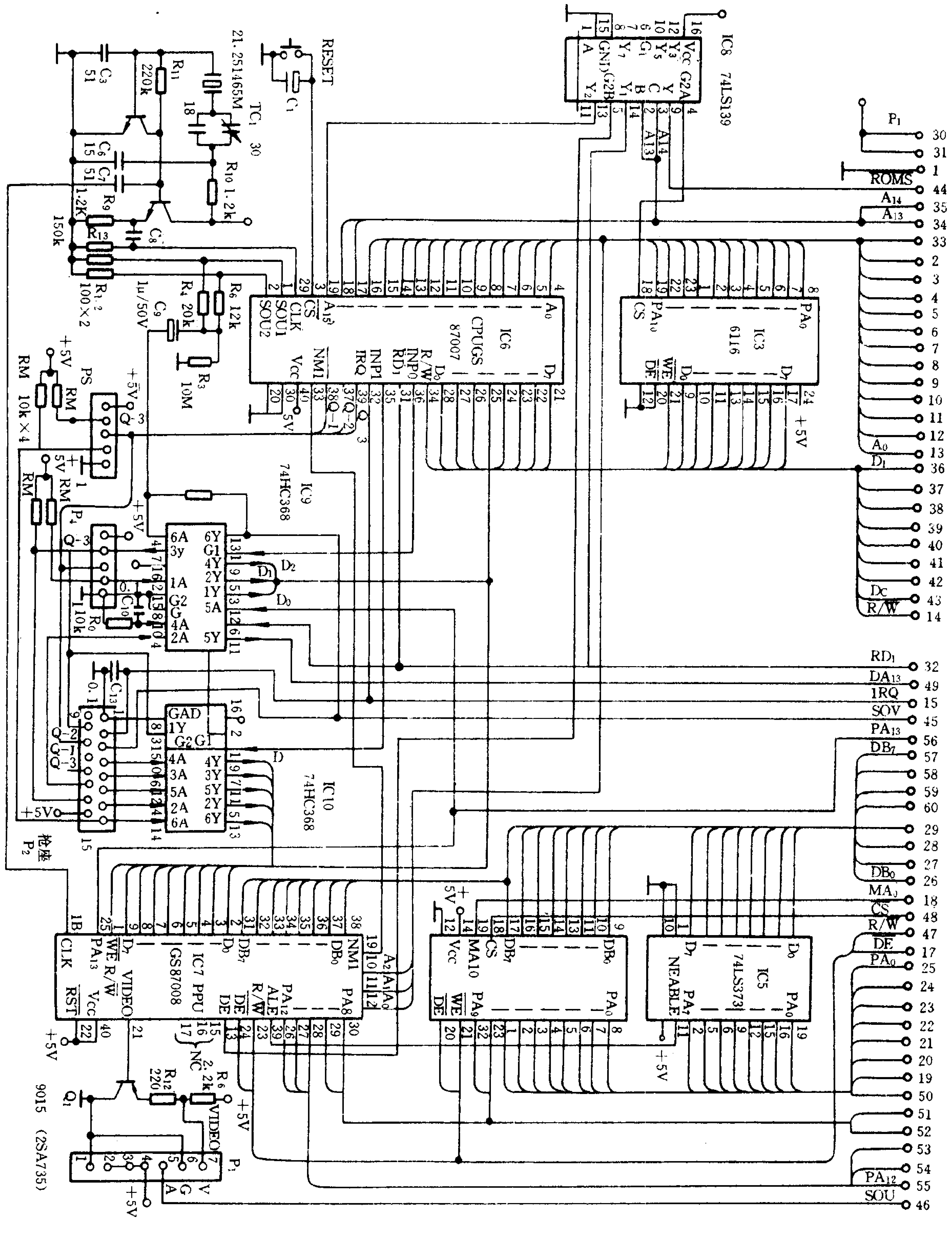


CT－180 调制板



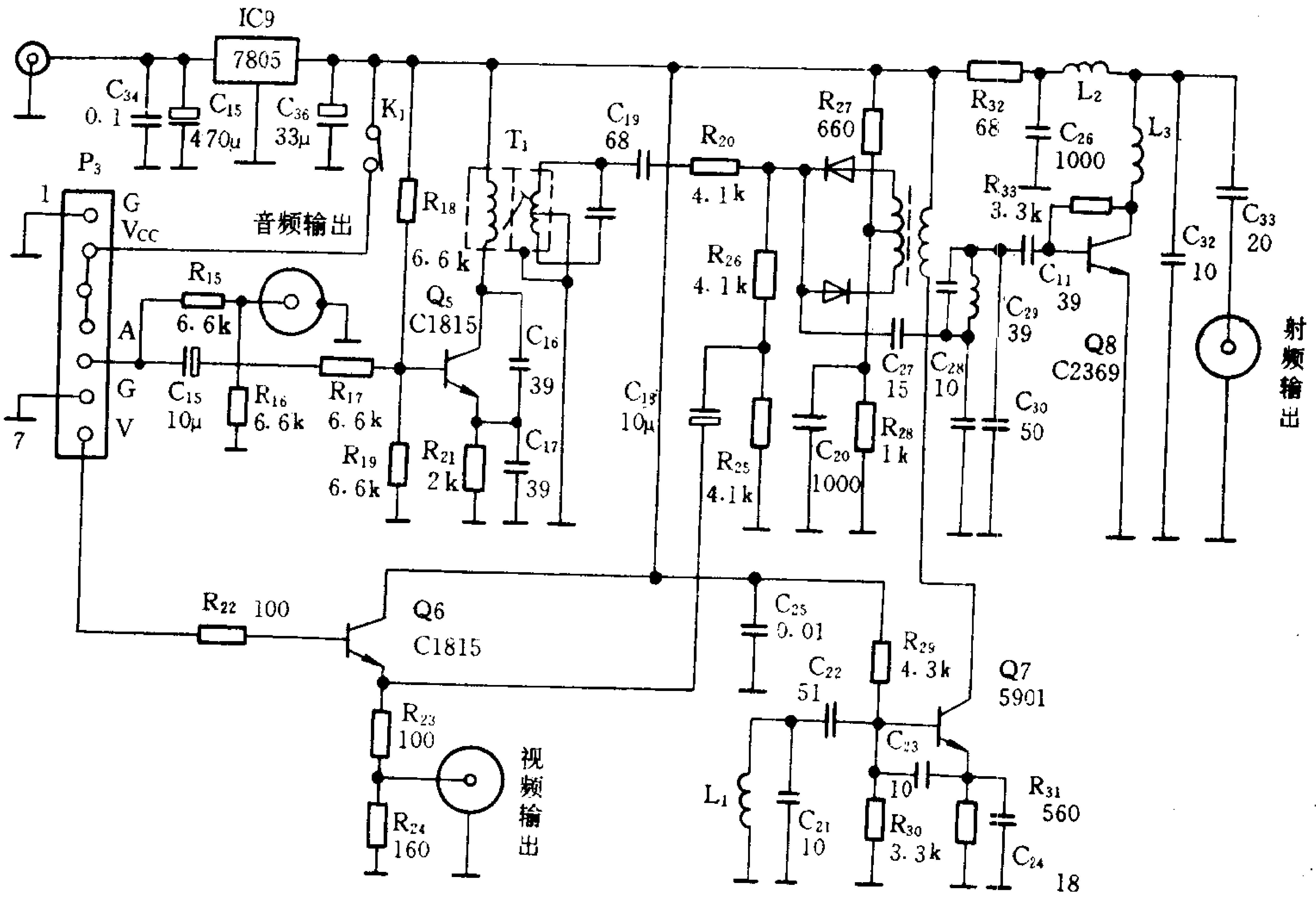


CT－180 手柄电路

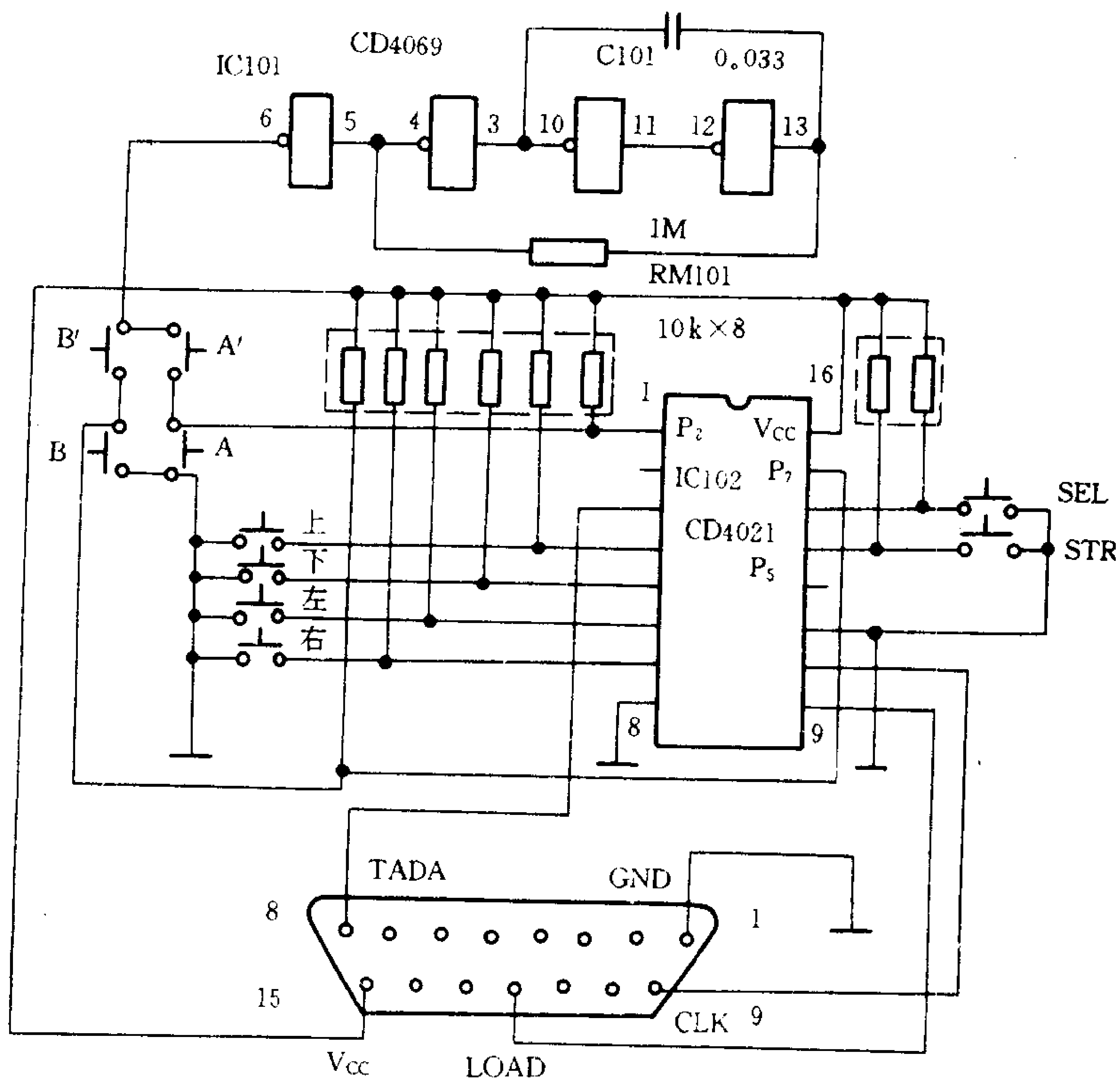


任天堂CT—180型游戏机电路原理图

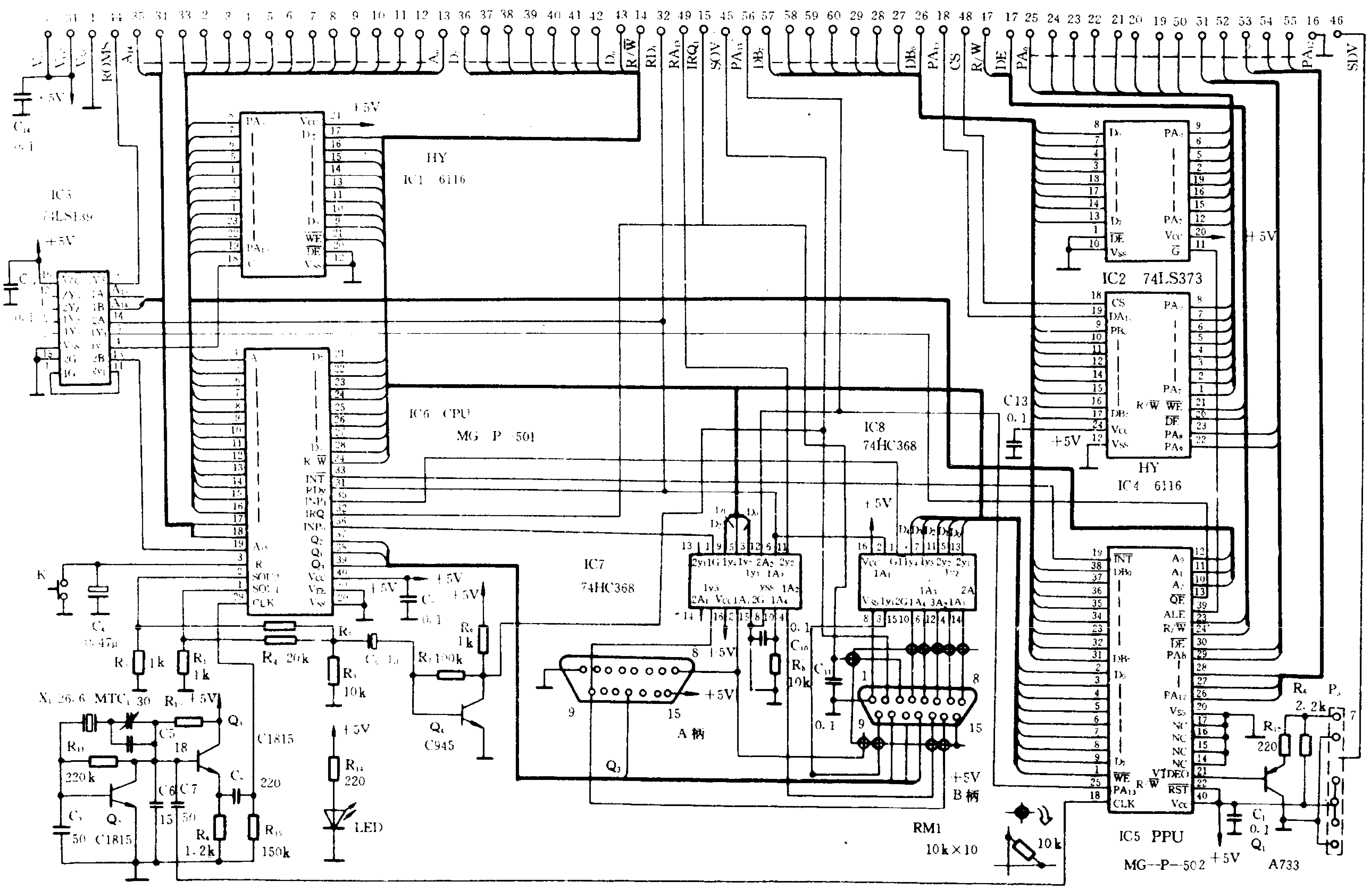
4. 小天才 501 电路原理图



小天才 501 调制板

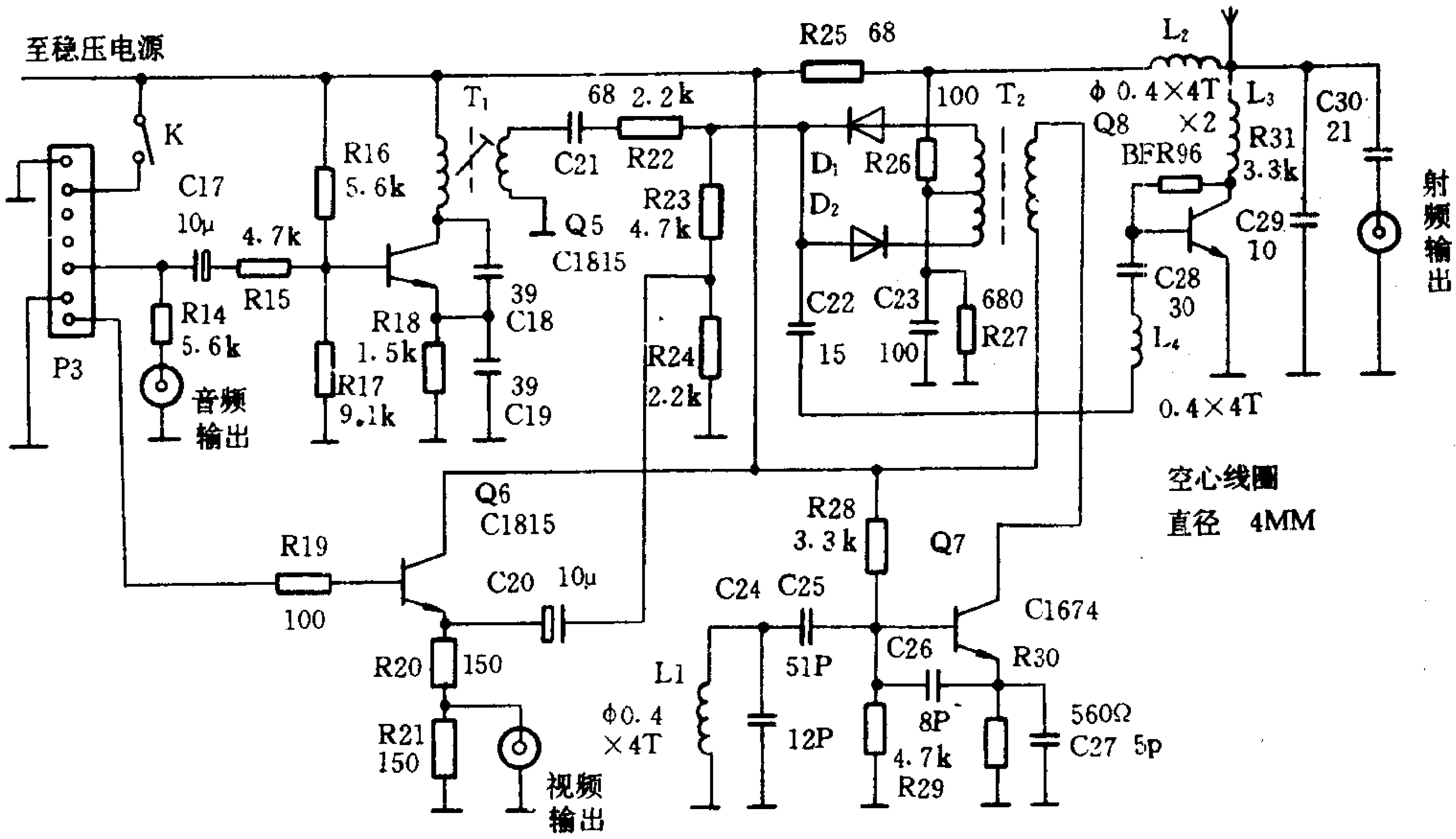


小天才 501 手柄

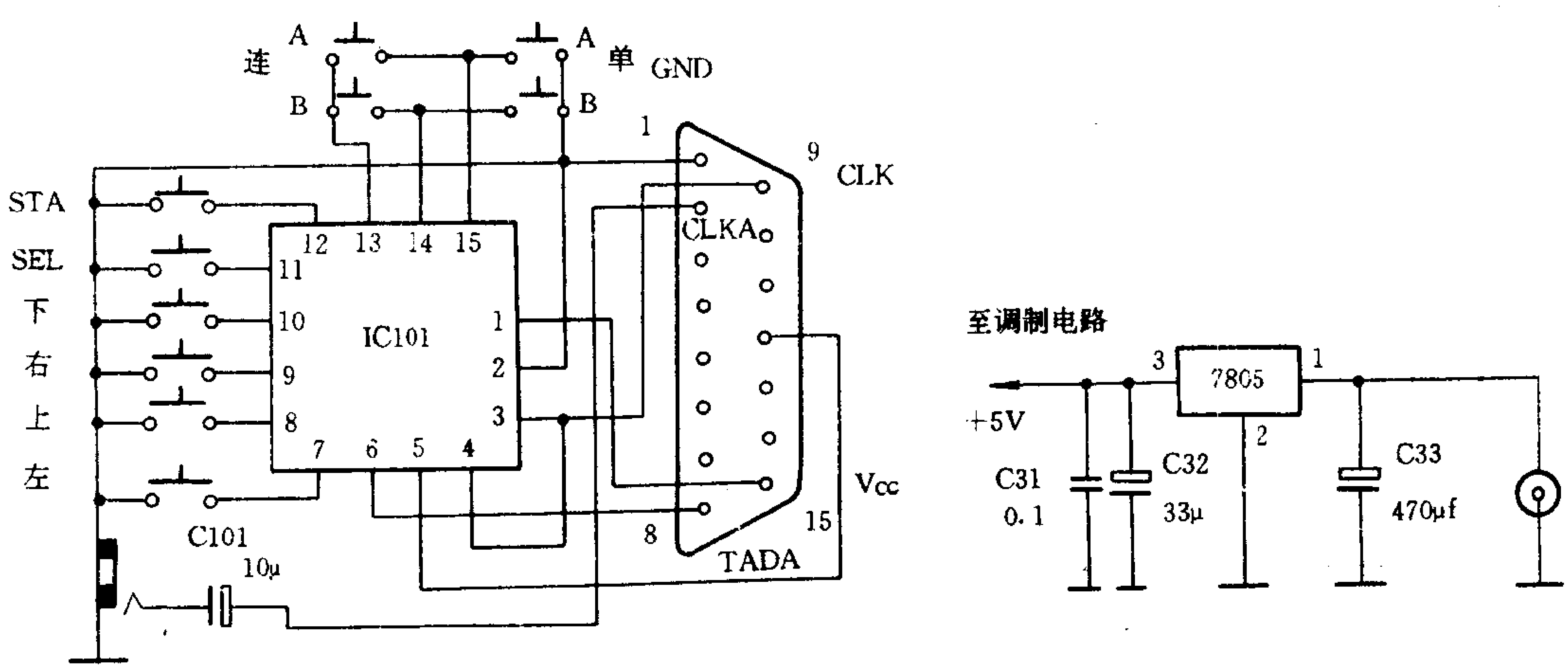


小天才 501 游戏机主机电路原理图

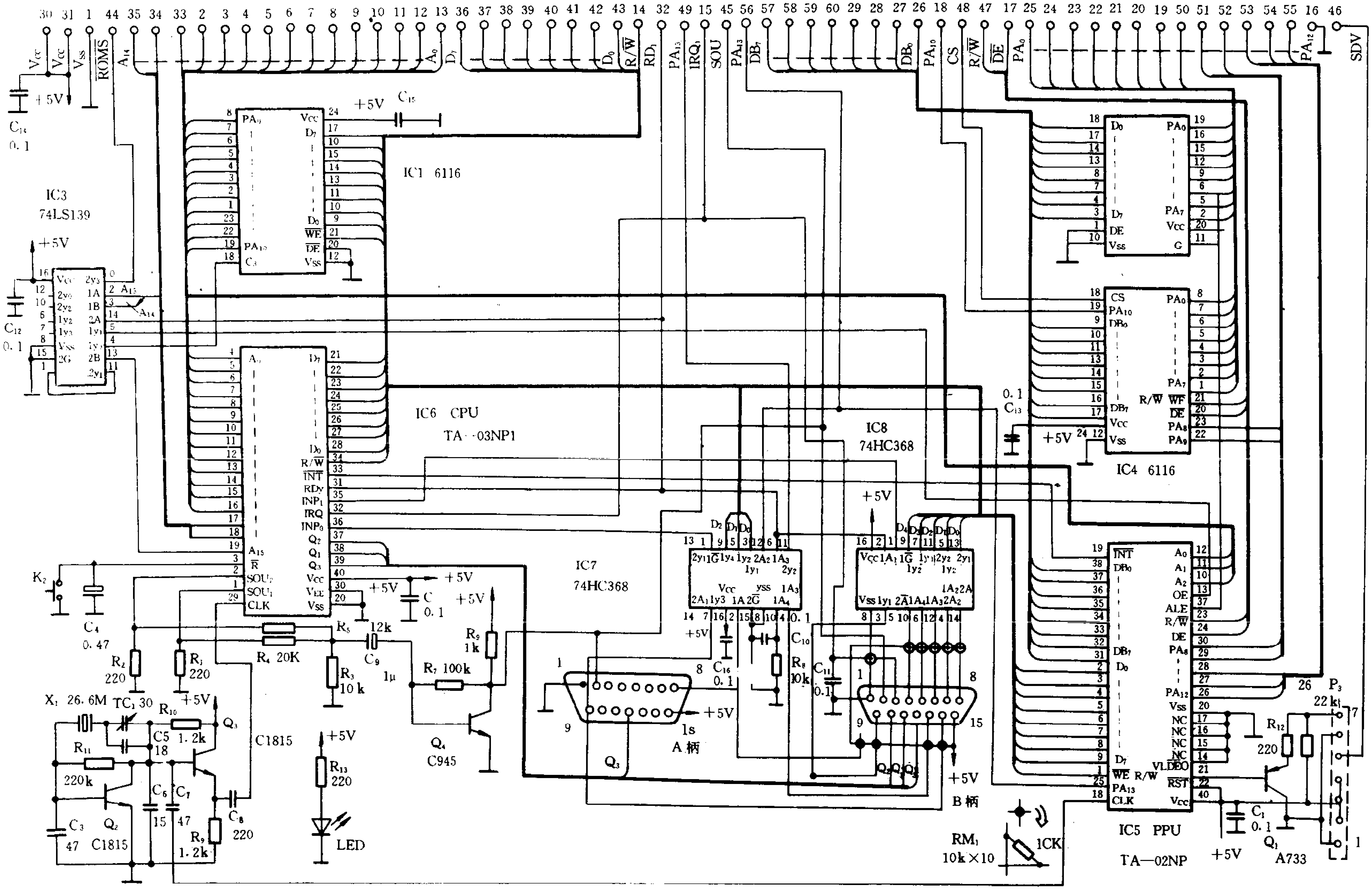
5. 终结者 S—500TM 电路原理图



S—500TM 调制板

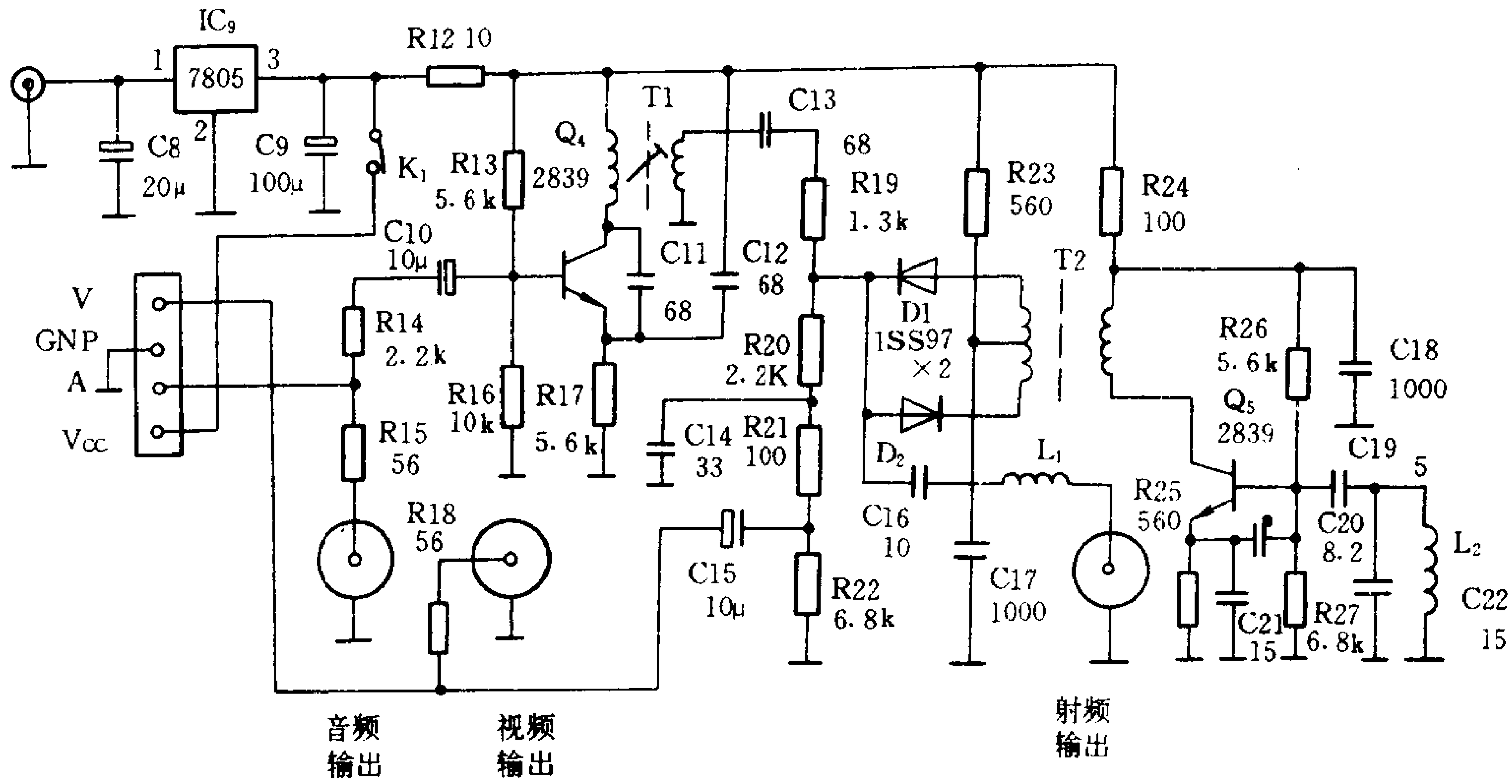


S—500TM 手柄及电源

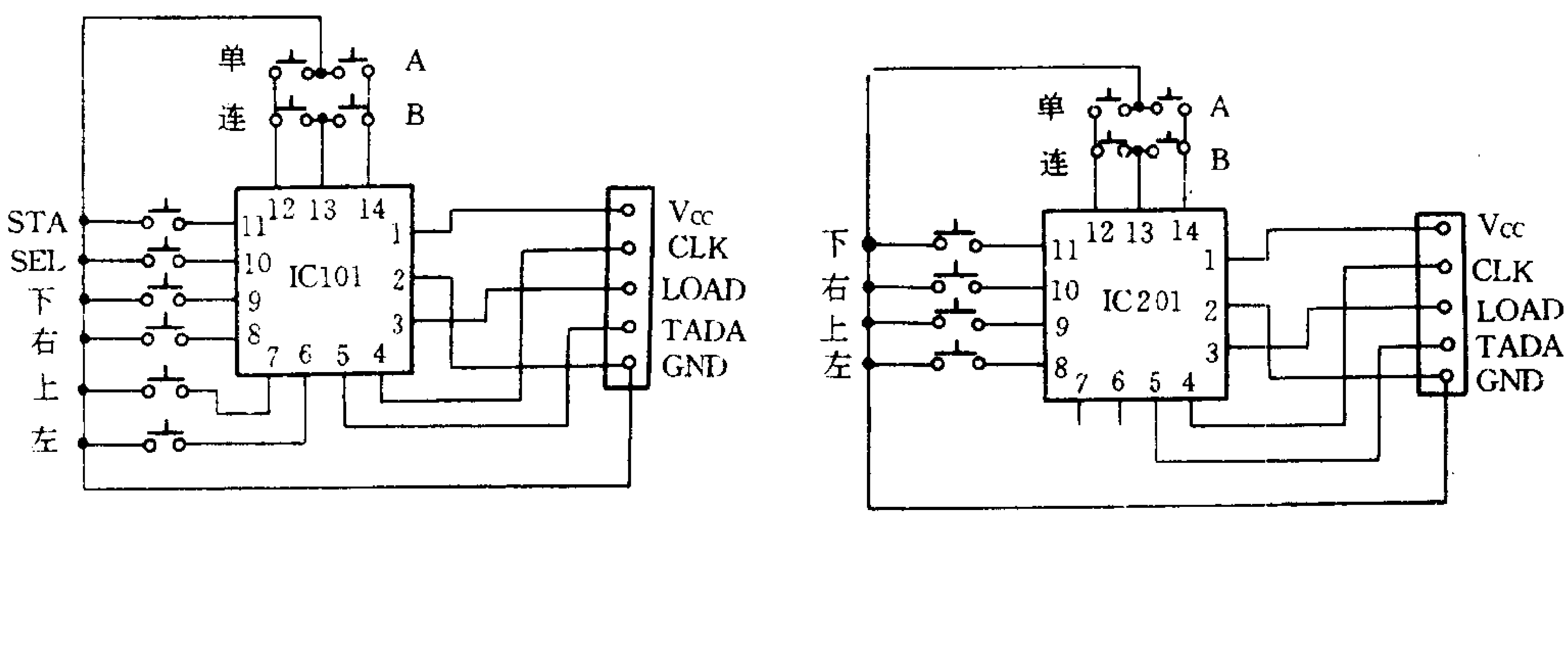


终结者S—500TM游戏机主机电路原理图

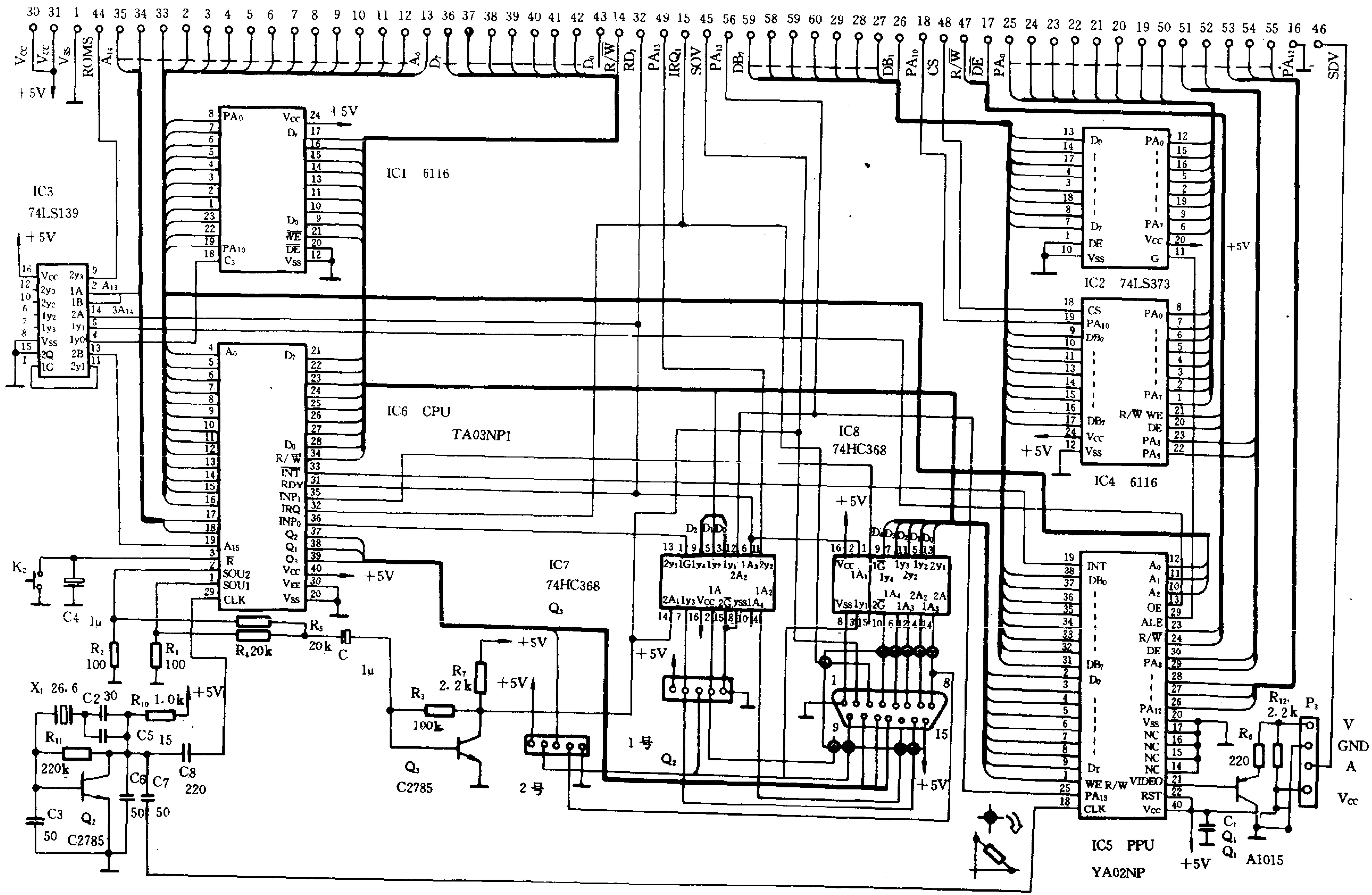
6. HTTEX HT767 电路原理图



HT767 调制板

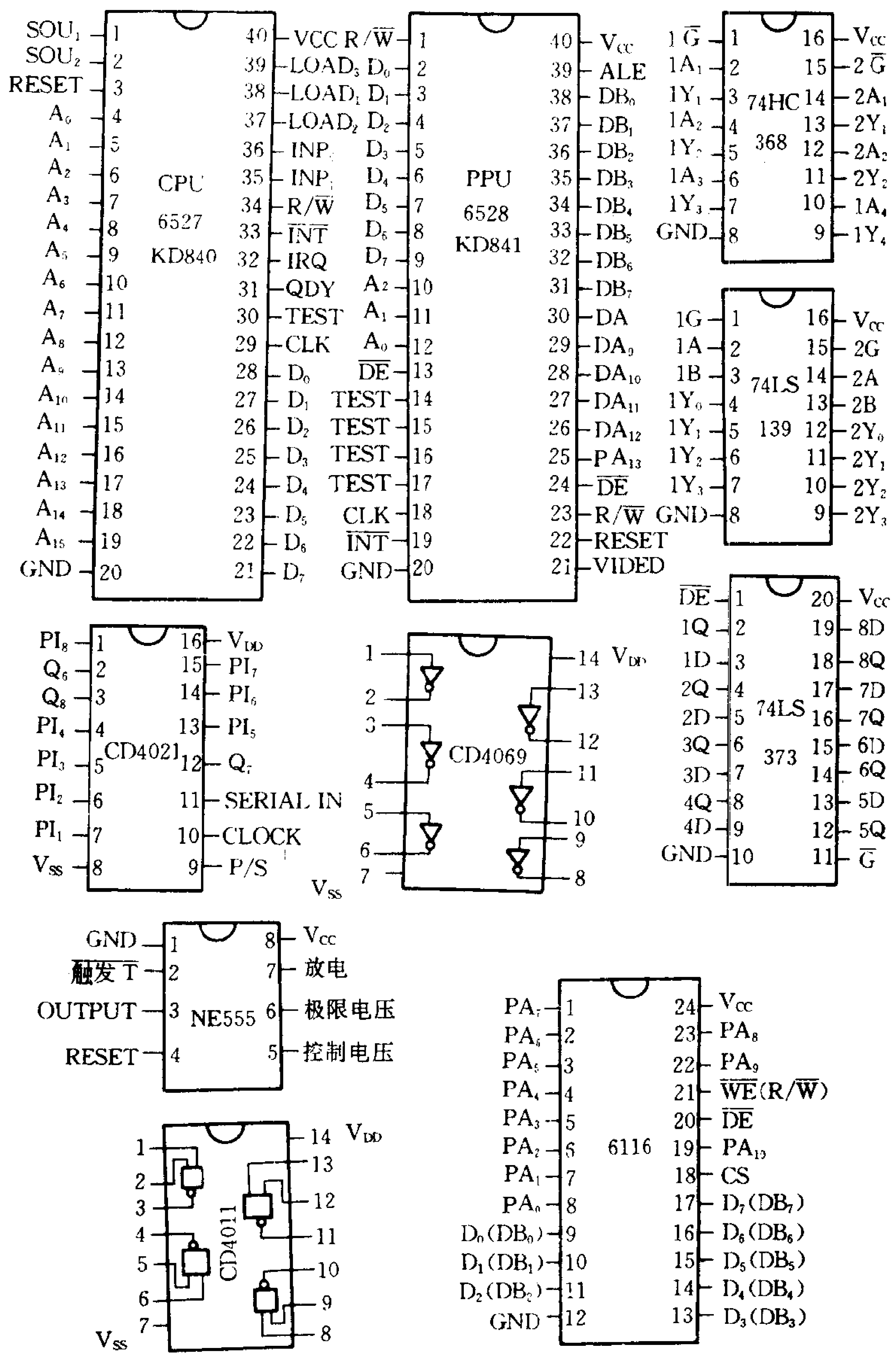


HT767 手柄



HT767 游戏机主机电路原理图

附录四　家用电视游戏机常用IC脚功能图



附录五　游戏机 IC 引脚阻值实测数据表

（此数据均用 500 型万用表在线测得）

CPU（中央处理器）　　　　（注：单位 kΩ，空格为无穷大）

引出脚	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	备 注
任天堂	0.2	0.2	10.5	8	7.8	8	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	4.1	4.1	4.1	0	黑地
CT-180	0.2	0.2	20																	0	红地
小天才	1	1	7.5	5.4	5.4	5.4	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5	5	5	0	黑地
IQ-301	1	1	23																	0	红地
小天才	1	1	13	11	11	11	11	11	11	11	11	11	11	11	11	11	10	10	10	0	黑地
IQ-501	1	1	23																	0	红地
终结者	0.2	0.2	6.7	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7	7	7	0	黑地
S-500TM	0.2	0.2	24																	0	红地
引出脚	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	备 注
任天堂	6.2	6.2	6.2	5.3	5.3	5	5	5	15	0	4.2	9.6	8	8	8	8.5	8.5	8.5	8	2.6	黑地
CT-180	10.5	10.5	10.5	10	10	9.5	9.5	9.5	200	0	11.5	12.5	12.5		11	11.5			11.5	2.5	红地
小天才	4.8	4.8	4.8	3.5	3.5	3.4	3.4	3.4	10	0	4.9	7.4	5.3	5.3	5.5	5.5	5.5	5.7	5.7	2.9	黑地
IQ-301	11.5	11.5	11.5	11.5	10	10	10	10	200	0	10.5	13	13		10.5	11.5			11	2.1	红地
小天才	11	11	11	9.5	9.5	9.5	9.5	9.5	18	0	10	11	11	11	11	11	11	11	11	3.5	黑地
IQ-501	20	20	20	18	18	18	18	18	200	0	23	13	13		21	24				3.5	红地
终结者	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	13.5	0	7	7	7	7.3	7.3	7.3	7.3	7.3	7.3	1.8	黑地
S-500TM	16	16	16	16	16	16	16	16		0	20	11	11		20	20				1.8	红地

PPU（视频处理器）　　　　（注：单位 kΩ，空格为无穷大）

引出脚	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	备 注
任天堂	8	5	5	5	5.3	5.3	5.3	6.2	6.2	7.8	7.8	7.8	5.9	0	0	0	0	6.5	8	0	黑地
CT-180		9.5	9.5	9.5	10	10	10.5	10.5	10.5				7.5	0	0	0	0	7.6	12.5	0	红地
小天才	5.4	3.4	3.4	3.4	3.6	3.6	4.8	4.8	4.8	5.4	5.4	5.4	6.7	0	0	0	0	11	5.4	0	黑地
IQ-301		10	10	10	10.2	10.2	11.5	11.5	11.5					0	0	0	0		13	0	红地
小天才	11	9.5	9.5	9.5	9.5	9.5	11	11	11	11	11	11	14	0	0	0	0	27	11	0	黑地
IQ-501		18	18	18	18	18	20	20	20					0	0	0	0		13	0	红地
终结者	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	15	0	0	0	0	25	7	0	黑地
S-500TM		16	16	16	16	16	16	16	16				24	0	0	0	0		10	0	红地
引出脚	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	备 注
任天堂	6	2.6	8.5	8.5	8.5	8.5	8.5	8.5	9.4	9.4	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.2	2.6	黑地
CT-180		2.6			11.5						11	11	11	11	11	11	11	11		2.5	红地
小天才	4.6	3	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	2.9	黑地
IQ-301		2.3			12															2.1	红地
小天才	1.3	3.7	17	17	17	17	17	17	17	17	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	3.5	黑地
IQ-501		3.8			2.4						22	22	22	22	22	22	22	22		3.5	红地
终结者	8.5	1.8	9.5	9.5	9.5	9.5	9.5	9.5	9.5	9.5	8	8	8	8	8	8	8	8	8	1..8	黑地
S-500TM		1.8			20						15	15	15	15	15	15	15	15		1.8	红地

6116(监控信号随机存储器)　　　　(注:单位 kΩ,空格为无穷大)

引出脚	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	备注
任天堂	8.5	8.5	8.5	8.5	8.5	8.9	7.8	7.8	5	5	5	0	5.2	5.2	6.2	6.2	6.2	6.2	8.5	0	8	8.5	8.5	2.6	黑地
CT-180									9.5	9.5	9.5	0	9.8	9.8	10.5	10.5	10.5	11.5		0				2.5	红地
小天才	5.4	5.4	5.4	5.7	5.7	5.7	5.7	5.7	4.8	4.8	4.8	0	3.6	3.6	3.4	3.4	3.4	7	5.8	0	5.4	5.8	5.8	2.9	黑地
IQ-301									11.5	11.5	11.5	0	10.2	10	10	9.8	10			0				2.3	红地
小天才	11	11	11	11	11	11	11	11	9.5	9.5	9.5	0	9.5	9.5	11	11	11	15	11	0	11	11	11	3.5	黑地
IQ-501									18	18	18	0	18	18	20	20	20			0				3.5	红地
终结者	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	0	7.5	7.5	7.5	7.5	7.5	14	7.3	0	7.3	7.3	7.3	1.8	黑地
S-500TM									15	15	15	0	15	15	15	15	15	24		0				1.8	红地

6116(视频信号随机存储器)　　　　(注:单位 kΩ,空格为无穷大)

引出脚	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	备注
任天堂	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	4.2	4.2	4.2	0	4.2	4.2	4.2	4.2	4.2	9.7	10	8.5	8.5	8.5	8.5	2.6	黑地
CT-180									10.8	10.8	10.8	0	10.8	10.8	10.8	10.8	10.8							2.5	红地
小天才	10	6.4	6.4	6.4	6.4	6.4	6.4	6.4	4.6	4.6	4.6	0	4.6	4.6	4.6	4.6	4.6	10	5.8	5.8	5.8	5.8	6.4	2.9	黑地
IQ-301												0												2.1	红地
小天才	2.3	13	13	13	13	13	13	13	8.5	8.5	8.5	0	8.5	8.5	8.5	8.5	8.5	23	17	17	17	17	13	3.5	黑地
IQ-501									22	22	22	0	22	22	22	22	22							3.5	红地
终结者	14	14	14	14	14	14	14	14	8	8	8	0	8	8	8	8	8	14	14	9.5	9.5	9.5	9.5	1.8	黑地
S-500TM									15	15	15	0	15	15	15	15	15							1.8	红地

74LS373(地址信号锁存器)　　(注:单位kΩ,空格为无穷大)

引出脚		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	备注
任天堂 CT-180		0	6.5	4.2	4.2	6.5	6.5	4.2	4.2	6.5	0	4.2	6.5	4.2	4.2	6.5	6.5	4.2	4.2	6.5	2.6	黑地
		0		10.8	10.8			10.8	10.8		0			10.8	10.8			10.8	10.8		2.5	红地
小天才 IQ-301		0	6.4	4.6	4.6	6.4	6.4	4.6	4.6	6.4	0	4.6	6.4	4.6	4.6	6.4	6.4	4.6	4.6	6.4	2.9	黑地
		0									0										2.1	红地
小天才 IQ-501		0	13	8.5	8.5	13	13	8.5	8.5	13	0	8.5	13	8.5	8.5	13	13	8.5	8.5	13	3.5	黑地
		0		22	22			22	22		0			22	22			22	22		3.5	红地
终结者 S-500TM		0	14	8	8	14	14	8	8	14	0	8	14	8	8	14	14	8	8	14	18	黑地
		0		15	15			15	15		0			15	15			15	15		1.8	红地

74LS139(译码器)　　(注:单位kΩ,空格为无穷大)

引出脚	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	备注
任天堂 CT-180	4.1	4.1	4.1	6.4	5.9	65.3	6.3	0	6.4	6.4	4.1	6.4	4.1	4.1	0	2.6	黑地
	12			11.5	7.5	12	12	0	12	12	12	11.5		11.5	0	2.5	红地
小天才 IQ-301	5.5	5	5	7	6.7	6.9	7	0	6.9	7	5.5	7	5	5	0	2.9	黑地
								0						10	0	2.1	红地
小天才 IQ-501	11	10	10	15	14	14	14	0	14	14	11	14	10	10	0	3.5	黑地
								0						23	0	3.5	红地
终结者 S-500TM	8.5	7.3	7.3	14	15	15	15	0	15	15	15	15	7	7	0	1.8	黑地
	24			24	24	24	24	0	24	24	24	24		20	0	1.8	红地

74HC368(总线驱动器主手柄)　　(注:单位kΩ,空格为无穷大)

引出脚	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	备注
任天堂 CT-180	8.5	7	5	8.5	5	4.2	5.5	0	5	7.7	5.5	8.5	5.5	10	0	2.6	黑地
	11.5	11.5	9.6	11.5	9.6	11.5	10	0	9.5	8	10.5	11.5	10.5	11.5	0	2.5	红地
小天才 IQ-301	5.6	6.7	3.4	7.6	3.4	5	3.6	0	3.4	7.2	3.8	6	3.6	8	0	2.9	黑地
	11	10.3	10	11.2	9.8	10.5	9.6	0	10	8	10	12	11.2	7.2	0	2.1	红地
小天才 IQ-501	11	13	9.5	12.5	9.5	10	10	0	9.5	10	11	17	11	17	0	3.5	黑地
	24	13	18	14	18	23	14	0	18	10	18.5	24	18.5	23	0	3.5	红地
终结者 S-500TM	7.3	11	7.3	11	7.3	7	9	0	7.3	10	10	9.5	10	17	0	1.8	黑地
	20	12	15	12	15	20	12	0	15	10	19	20	19	24	0	1.8	红地

74HC368(总线驱动器,副主柄)　　　　(注:单位kΩ,空格为无穷大)

引出脚	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	备注
任天堂 CT-180	8.5	4.2	5.4	8.5	5	8.5	5.5	0	5.4	8.5	5	8.5	5	6.7	8.2	2.6	黑地
	11	11.5	10	11.2	9.6	11.2	10	0	10	11.2	9.6	11.2	9.6	10	11	2.5	红地
小天才 IQ-301	5.6	4.9	3.6	7.6	3.4	7.6	3.6	0	3.4	7.6	3.4	7.6	3.4	6.8	5.6	2.9	黑地
	10.8	10.8	10.8	11.5	10	11	10	0	10.5	11.5	10	11.2	10	10.5	10.5	2.1	红地
小天才 IQ-501	11	10	10	12.5	9.5	12.5	9.5	0	9.5	12.5	9.5	12.5	9.5	12.5	11	3.5	黑地
	21	23	23	13	18	13	18	0	18	13	18	13	18	13	21	3.5	红地
终结者 S-500TM	7.3	7	9	11	7.3	11	7.3	0	7.3	11	7.3	11	7.3	11	7.3	1.8	黑地
	20	20	12	11	14	12	12	0	14	12	14	12	14	12	20	1.8	红地

附录六　电视游戏机IC代换表

型号	功能	代换型号
6526	视频处理器	C11903　CO11903
6527AP	中央处理器	6005　P03　KD840　2A03E　6540　87007　87007P
VA6527	21M 中央处理器	6005B，2A03E，TA03N，GS87007，KD840，6540，H27P02 (HH41)
VA6527P	26M 中央处理器	GS8007P　P02(HI21)、MG－P－501、YA03NP1、0916(9041)、KD840P
VA6528	NTSC 制视频处理器	6022B、2C02、TA02N、GS87008、KD841、6541、H27P03(HH25)(HH23)'　2D03
VA6528P	PAL 制视频处理器	GS8008P、P03－1(HI15)、MG－P－502、TA02NP、0615B(9035)、KD841P、VA6538、MG－P－505、2D02
6116	随机存储器	GM7628－10、F3528、AM9128、MCM2016、MS8725、NMC2116、SY2128、SY2129、TMM2015A、TMM2016A、MPTD4016
74LS139	译码器	CT4139、HD74LS139、T74LS139、SN74LS139、DN74LS139
74LS373	锁存器	SN74LS373、HD74LS373;T74L373、DN74LS139
74HC368	六反相器总线驱动器	CC74HC368、MC74HC368、CD74HC368、MM74HC368、SP74HC368
4021	八位移位寄存器	CC4021、DG4021B、HD14021、MC14021、F4021、CD4021、TC4021
4069	六反相总线驱动器	F4069、HD14069、MC14069、C033、CD4069、TC4069、MM54C04、MM74C04
MK5060	制式转换器	NK5060、MK5060、NPAL5/6
502	视频处理器(PAL 制)	6538P、6538AP、6528P、P02、87008P、2D02
501	中央处理器	6527P